μP25

ALB E25 - ALB S25

microProcessor 25 I/O lines

MANUALE TECNICO

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@grifo.it
FAX +39 051 89 36 61
Tel. +39 051 89 20 52 (4 lin.r.a.)

μP25, ALB E25, ALB S25 Edizione 3.1 *Rel. 18 Ottobre 1996*

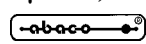
µP25

ALB E25 - ALB S25

microProcessor 25 I/O lines

MANUALE TECNICO

µP25
Dispositivo basato sul microcontrollore 87c51, sul quale é stato implementato un apposito firware di gestione. Tramite sequenze di codici sulla linea seriale, é possibile: comandare 3 port da 8 bit per operazione di I/O, generando eventualmente segnali temporizzati; acquisire il counter a 16 bit; leggere o scrivere messaggi dalla EEPROM; gestire con comandi ad alto livello, un A/D, un D/A ed un LED display driver. Si possonono selezionare due protocolli di comanicazione, per collegamenti punto-punto o in rete con una struttura Master-Slave.
Modalità di SETUP locale per la completa configurazione delle varie sezioni.

ALB E25
Scheda valutativa basata sul **µP25**, provvista di una serie di LEDs, tasti e connettori, che permettono all'utente di verificare tutte le varie risorse di cui il **µP25** é provvisto. Linea seriale configurabile in RS232 o RS485.

ALB S25
Scheda sperimentale basata sul **µP25**, provvista di un'area prototipale, in cui può essere sviluppato un apposito hardware per l'interfacciamento e la specializzazione delle linee di I/O del **µP25**. Linea seriale configurabile in RS232 o RS485.

Via dell' Artigiano, 8/6
40016 San Giorgio di Piano
(Bologna) ITALY
Email: grifo@grifo.it
FAX +39 051 89 36 61
Tel. +39 051 89 20 52 (4 lin.r.a.)

µP25, ALB E25, ALB S25 Edizione 3.1 *Rel. 18 Ottobre 1996*

IMPORTANTE

Tutte le informazioni contenute in questo manuale sono state accuratamente verificate, ciononostante **grifo**® non si assume nessuna responsabilità per danni diretti o indiretti a cose e/o persone derivanti da errori tecnici ed omissioni o dall'uso del presente manuale, del software o dell'hardware ad esso associato.
grifo® altresi si riserva il diritto di modificare il contenuto e la veste di questo manuale senza alcun preavviso, con l'intento di offrire un prodotto sempre migliore, senza che questo rappresenti un obbligo **grifo**®.
Per le informazioni specifiche sui componenti montati sulla scheda, l'utente deve fare riferimento ai Data Book delle case costruttrici o delle seconde sorgenti.

INDICE GENERALE

INDICE DELLE FIGURE

INTRODUZIONE

L’uso di questi dispositivi é rivolto - IN VIA ESCLUSIVA - a personale specializzato.

Scopo di questo manuale é la trasmissione delle informazioni necessarie all’uso competente e sicuro del prodotto. Esse sono il frutto di un’elaborazione continua e sistematica di dati e prove tecniche reistrate e validate dal Costruttore, in attuazione alle procedure interne di sicurezza e qualità dell’informazione.

I dati di seguito riportati sono destinati - IN VIA ESCLUSIVA - ad un utenza specializzata, in grado di interagire con il prodotto in condizioni di sicurezza per le persone, per la macchina e per l’ambiente, interpretando un’elementare diagnostica dei guasti e delle condizioni di funzionamento anomale e compiendo semplici operazioni di verifica funzionale, nel pieno rispetto delle norme di sicurezza e salute vigenti.

Le informazioni riguardanti installazione, montaggio, smontaggio, manutenzione, aggiustaggio, riparazione ed installazione di eventuali accessori, dispositivi ed attrezzature, sono destinate - e quindi eseguibili - sempre ed in via esclusiva da personale specializzato avvertito ed istruito, o direttamente dall’ASSISTENZA TECNICA AUTORIZZATA, nel pieno rispetto delle raccomandazioni trasmesse dal costruttore e delle norme di sicurezza e salute vigenti.

Per un corretto rapporto col prodotto, é necessario garantire leggibilità e conservazione del manuale, anche per futuri riferimenti.
In caso di deterioramento o più semplicemente per ragioni di approfondimento tecnico ed operativo, consultare direttamente l’Assistenza Tecnica autorizzata.

Al fine di non incontrare problemi nell’uso di tali dispositivi, é conveniente che l’utente - PRIMA DI COMINCIARE AD OPERARE - legga con attenzione tutte le informazioni contenute in questo manuale.
In una seconda fase, per rintracciare più facilmente le informazioni necessarie, si può fare riferimento all’indice generale e all’indice analitico, posti rispettivamente all’inizio ed alla fine del manuale.

DESCRIZIONE GENERALE DEL µP25

Il **µP25** (microProcessore **25 linee I/O)**, é un potente sistema, appositamente realizzato per tutte le applicazioni, nelle quali é richiesta la gestione di linee in Input o Output.
Tale dispositivo é basato sul diffusissimo microcontrollore **87c51**, sul quale é stato implementato un apposito firmware di gestione; in questo modo, tramite comodi comandi sulla linea seriale, é possibilegestire varie sezioni. In particolare sono disponilili **3 Port a 8 bit** per operazioni di I/O, quali il controllo di linee in Output, eventualmente generando dei segnali temporizzati (onda quadra, settaggio o resettaggio del pin per un determinato tempo, ecc.), acquisizione di linee in Input eventualmente con debauncing; ecc.
E' inoltre disponibile una ulteriore linea, che può essere settata per funzionare in I/O, oppure come segnale di trigger, per il **Counter a 16 bit**, che viene mosso a disposizione dell'utente.
Per poter funzionare correttamente, il **µP25** deve essere collegato ad una **EEPROM** seriale, con capacità fino a **2KBytes**; questa oltre a contenere tutte le varie configurazioni del sistema, é utilizzata per memorizzare fino a **203 stringhe** di 10 caratteri ciascuna, sempre utilizzabili dall'utente.
Il **µP25** mette poi a disposizione una serie di comandi, per la gestione ad alto livello, di un **A/D converter**, di un **D/A converter** e di un **LED display driver**, opportunamente intefacciati tramite alcune delle linee di I/O.
Sono disponibili due protocolli di comunicazione con Baud Rate settabile fino a **38.4 KBaud**; il primo da utilizzare nel caso in cui si voglia realizzare normale collegamento punto-punto, in **RS232**, **RS422** o **Current Loop**; l'altro necessario per sviluppare una struttura **Master-Slave**, con collegamento in **RS485** o **Current Loop**. In quest'ultimo caso, collegando più dispositivi **µP25** in rete, é possibile gestire un numero molto elevato di linee di I/O, anche a notevole distanza tra di loro, sfruttando un'unica struttura Master ed una sola linea seriale; ad esempio possono essere realizzati controlli di porte automatiche per edifici, ripetori di segnali, ecc.

- Implementazione firmware su microcontrollore **87c51**.
- Modalità di SETUP locale per la completa configurazione del sistema.
- Protolocollo di comunicazione per sistemi punto-punto o **Master-Slave**.
- Fino a **128** dispositivi **µP25** collegabili in rete in modalità Master-Slave.
- Baud Rate selezionabile da **1200 Baud** a **38.4 KBaud**.
- Fino a **25 linee** gestibili per operazioni di I/O
- Possibilità di generare segnali quali onda quadra, settaggio o resettaggio temporizzato di un pin,ecc., il tutto gestito ad alto livello.
- Supporta varie **EEPROM** seriali fino a **2KBytes**, per il salvataggio di Setup e messaggi.
- Fino a **203 stringhe** di 10 caratteri memorizzabili nella EEPROM.
- **Counter a 16 bit** gestito ad alto livello, a disposizione utente.
- **A/D converter a 12 bit TLC 2543** intefacciabile e comandabile ad alto livello.
- **D/A converter a 16 bit AD 420** interfacciabile e comandabile ad alto livello.
- **LED display driver M5480** interfacciabile e comandabile ad alto livello.

DESCRIZIONE HARDWARE DEL μP25

Viene riportata di seguito una descrizione hardware delle varie sezioni che compongono un sistema minimo basato sul **μP25**, (il cui schema elettrico é riportato in figura 1), con indicate le varie funzioni svolte da ciascuno di essi.

LINEE DI INPUT/OUTPUT DEL PORT 0

E' composto dalle 8 linee denominate **P0.?** (pin 36÷43 del **μP25**), le quali sono configurabili singolarmente come pin di Input o Output. I comandi disponibili per tale sezione permettono operazioni di sia a livello di byte (scrittura o lettura dell'intero port), che di bit (scrittura o lettura di una singola linea).
Inoltre, una serie di comandi, non disponibili per i **Port 1** e **2**, permette di eseguire delle **variazioni di stato temporizzate** sulle linee configurate in Output; infatti é possibile generare un'onda quadra, attivare una determinata uscita per un certo tempo, generare un numero determinato di impulsi, ecc.

LINEE DI INPUT/OUTPUT DEL PORT 1

E' composto dalle 8 linee denominate **P1.?** (pin 2÷9 del **μP25**), le quali sono configurabili singolarmente come pin di Input o Output. I comandi disponibili per tale sezione permettono operazioni di sia a livello di byte (scrittura o lettura dell'intero port), che di bit (scrittura o lettura di una singola linea).

LINEE DI INPUT/OUTPUT DEL PORT 2

E' composto dalle 8 linee denominate **P2.?** (pin 24÷31 del **μP25**), le quali possono essere configurate singolarmente come pin di Input, Output, oppure segnali di gestione di **A/D converter**, **D/A converter** o **LED display driver** (come sarà descritto in seguito).
Nel caso di configurazione come pin di I/O, i comandi disponibili per tale sezione permettono operazioni di sia a livello di byte (scrittura o lettura dell'intero port), che di bit (scrittura o lettura di una singola linea).

LINEA INT0

Corrisponde al segnale **/INT0** (pin 14 del **μP25**), il quale può essere configurato come pin di Input, Output, oppure come segnale di trigger per il **Counter a 16 bit**, a disposizione dell'utente.
In questo caso particolare, il contatore in questione, sarà incrementato di una unità in corrispondenza ad ogni **fronte di discesa**, del segnale applicato a tale pin.
Se, invece, la linea **INT0** é configurata come I/O, oltre ai comandi per il settaggio o l'acquisizione del pin, sono disponibili le istruzioni che permettono di eseguire le variazioni di stato temporizzate, descritte nel paragrafo relativo al Port 0.

LINEA DI RESET

Corrisponde al segnale **RST** (pin **10** del **μP25**), il quale deve essere collegato ad una apposita circuiteria che si occupa di generare un segnale di **Reset** durante la fase di Power-ON del sistema. Tale segnale, in questa fase, deve andare a livello logico **1**, restare in questo stato per un tempo minimo di **5 μsec**, quindi tornare a livello locico **0**.

SEZIONE DI SELEZIONE MODO E ALIMENTAZIONE RETI RESISTIVE

Corrisponde alla circuiteria pilotata dal segnale **/SETUP** (pin **17** del **μP25**), la quale ha un doppio funzionamento: tramite il **Jumper** si seleziona il modo di funzionamento (**RUN o SETUP**) , mentre con il transiston **PNP**, viene gestita l'alimetazione delle reti resistive (segnale **SIP**) dei Port **0**, **1** e **2**, del **μP25**.
In particolare si possono verificare due condizioni:

Jumper connesso (modo SETUP):
In questa fase, il **μP25** setta tutti i port in Input, e le relative reti resistive risultano automaticamente alimentate, in quanto, il transistor viene tenuto in conduzione dal jumper stesso.

Jumper non connesso (modo RUN):
In questa fase, il **μP25** verifica, innanzittutto, i dati di settaggio contenuti nella EEPROM; se questi sono validi vengono configurati i Port in Input o in Output e, solo a questo punto, viene fornita alimentazione alle reti resistive (LED acceso). In questo modo, durante la fase di inizializzazione, seguente ad un Power-ON, é garantito lo stato logico **0** ai pin che sono configurati in Output, evitando quindi l'attivazione accindetale di dispositivi di tipo **NPN** ad essi collegati.
Se, invece, i dati contenuti nella EEPROM non sono validi, i port rimangono in Input, e le reti resistive non vengono alimentate (LED spento).

SEZIONE DI GESTIONE EEPROM

E' la circuiteria formata dalla **EEPROM** seriale e dal relativo interfacciamento. Questa sezione viene pilotata dai segnali **PWR**, **SCL** e **SDA** (pin **15**, **18** e **19** del **μP25**), in particolare:

PWR (pin 15) -> Linea di controllo della circuiteria di alimentazione dell'EEPROM
SCL (pin 18) -> Linea per segnale seriale di **Clock**
SDA (pin 19) -> Linea per segnale seriale di **Data**

La circuiteria che gestisce l'alimentazione dell'EEPROM garantisce un migliore mantenimento dei dati in essa memorizzati, in quanto questa risulterà alimentata solo nella fasi di lettura e scrittura dielle informazioni. Tale circuiteria é comunque opzionale, e la si può omettere lasciando il pin **PWR** del **μP25** scollegato.

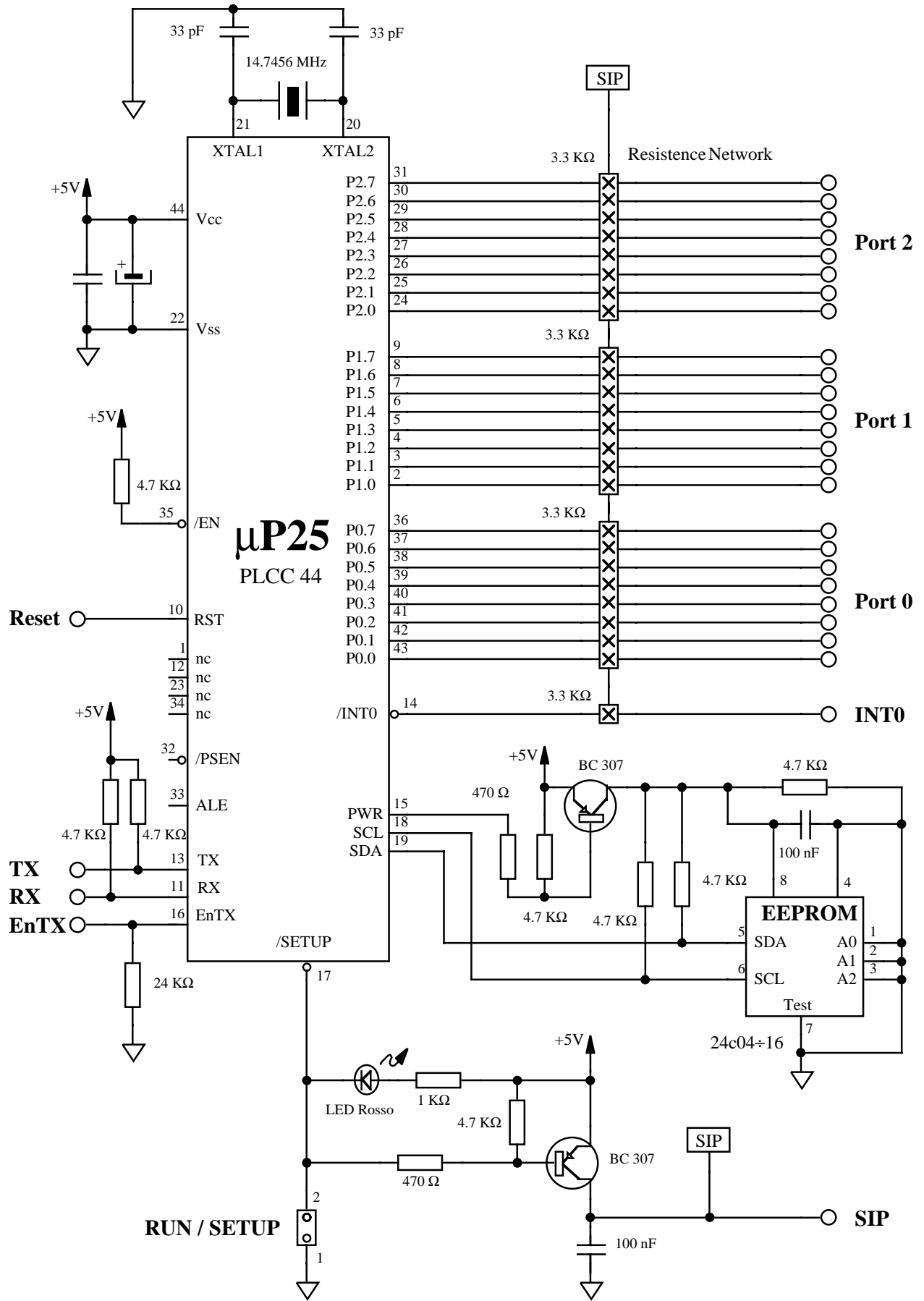



FIGURA 1: SCHEMA ELETTRICO DEL SISTEMA MINIMO BASATO SUL µP25

INTERFACCIAMENTO DI A/D, D/A E LED DISPLAY DRIVER

Tramite le linee del Port **2**, é possibile collegare al **µP25**, un **A/D converter**, un **D/A converter** ed un **LED display driver**. Questi dispositivi, vengono gestiti tramite una serie di comandi ad alto livello, e permettono di rendere la struttura basata sul **µP25** ancora più potente e versative, rendendo ancora più ampio il ventaglio di applicazioni nelle quali può essere usata.
Le modalità di collegamento di tali periferiche al port **2**, sono illustrate nella figura seguente.

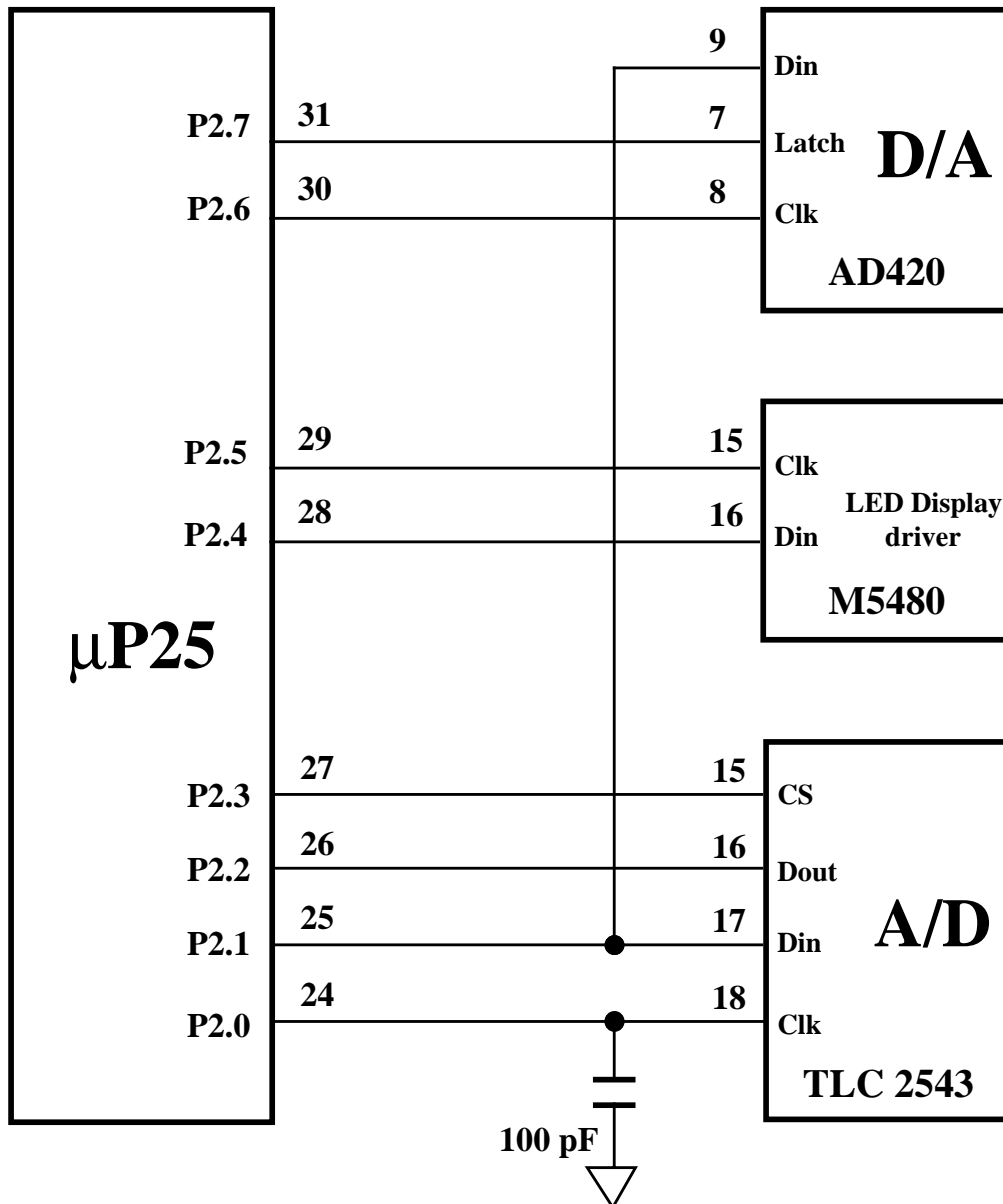



FIGURA 2: INTERFACCIAMENTO DI A/D, D/A E LED DISPLAY DRIVER AL µP25

INTERFACCIAMENTO DELL'A/D CONVERTER

Al **µP25**, può venire collegato l'A/D converter **TLC 2543** della **Texas Instruments**; questo dispositivo ha le seguenti caratteristiche:

- Conversione segnale con risoluzione a **12 bit**.
- **11** canali analogici d'ingresso.
- **10 µsec** di tempo di conversione.
- Pin di output di fine conversione (**E**nd **O**f **C**onversion).

Per interfacciare il **TLC 2543** al **µP25**, é necessario effettuare i seguenti collegamenti:

Port 2.0 (pin **24** del **µP25**)	-> **Clk** (pin **18** del **TLC 2543**)	: Segnale seriale di Clock
Port 2.1 (pin **25** del **µP25**)	-> **Din** (pin **17** del **TLC 2543**)	: Segnale seriale di Data In
Port 2.2 (pin **26** del **µP25**)	<- **Dout** (pin **16** del **TLC 2543**)	: Segnale seriale di Data Out
Port 2.3 (pin **27** del **µP25**)	-> **CS** (pin **15** del **TLC 2543**)	: Segnale di Cheap Select

INTERFACCIAMENTO DEL D/A CONVERTER

Al **µP25**, può venire collegato il D/A converter **AD 420** dell'**Analog Device**; questo dispositivo ha le seguenti caratteristiche:

- Conversione segnale con risoluzione a **16 bit**.
- Segnale d'uscita selezionabile in **Tensione** o **Corrente**, nei range: **0÷10 V**, **4÷20 mA**, **0÷20 mA** e **0÷24 mA**.

Per interfacciare l'**AD420** al **µP25**, é necessario effettuare i seguenti collegamenti:

Port 2.1 (pin **25** del **µP25**)	-> **Din** (pin **9** dell'**AD420**)	: Segnale seriale di Data In
Port 2.6 (pin **30** del **µP25**)	-> **Clk** (pin **8** dell'**AD420**)	: Segnale seriale di Clock
Port 2.7 (pin **31** del **µP25**)	-> **Latch** (pin **7** dell'**AD420**)	: Segnale di settaggio dell'uscita

INTERFACCIAMENTO DEL LED DISPLAY DRIVER

Al **µP25**, può venire collegato il LED display driver, denominato **M5480** da **SGS-Thomson,** oppure **MM5480** da **National Semiconductor**; questo dispositivo ha le seguenti caratteristiche:

- **23** uscite con driver generatori di corrente
- Possibilità di comandare direttamente un display a LED a **3 cifre e mezzo**.
- Possibilità di collegare dei LED direttamente alle uscite, senza resistenze di limitazione.

Per interfacciare il **M5480** al **µP25**, é necessario effettuare i seguenti collegamenti:

Port 2.4 (pin **28** del **µP25**)	-> **Din** (pin **16** del **M5480**)	: Segnale seriale di Data In
Port 2.5 (pin **29** del **µP25**)	-> **Clk** (pin **15** del **M5480**)	: Segnale seriale di Clock

SEZIONE PER LA COMUNICAZIONE SERIALE

E' composta dai segnali **TX**, **RX** e **EnTX** (pin **13**,**11** e **16** del **µP25**), i quali, opportunamente interfacciati, permettono di effettuare il collegamento seriale tra l'unità master ed il sistema basato sul **µP25**. Questo può essere in **RS232**, **RS422** o **Current-Loop**, se si utilizza la modalità di comunicazione punto-punto (unità master collegata ad un solo **µP25**), oppure in **RS485** o **Current-Loop**, nel caso di comunicazione in Master-Slave.
I pin del **µP25** in questione, hanno il seguente significato:

TX **(pin 13 del µP25)** -> Linea di trasmissione seriale.
RX **(pin 11 del µP25)** -> Linea di ricezione seriale.
EnTX **(pin 16 del µP25)** -> Segnale per l'abilitazione del dispositivo di trasmissione:
*Livello logico **0**:* Trasmettitore **disattivo**.
*Livello logico **1**:* Trasmettitore **attivo**.

L'utilizzo di quest'ultima linea é necessario, solo nel caso in cui si voglia realizzare un interfacciamento in **RS485**, per un collegamento in rete con il protocollo Master-Slave. Con tale modalità di comunicazione, infatti, il dispositivo di trasmissione deve essere attivato solo quando il **µP25** deve trasmettere delle informazioni all'unità Master.
In questo caso, può essere realizzata la circuiteria di interfacciamento, basata sul **SN75176** della **Texas Instruments** e riportata nella figura seguente.

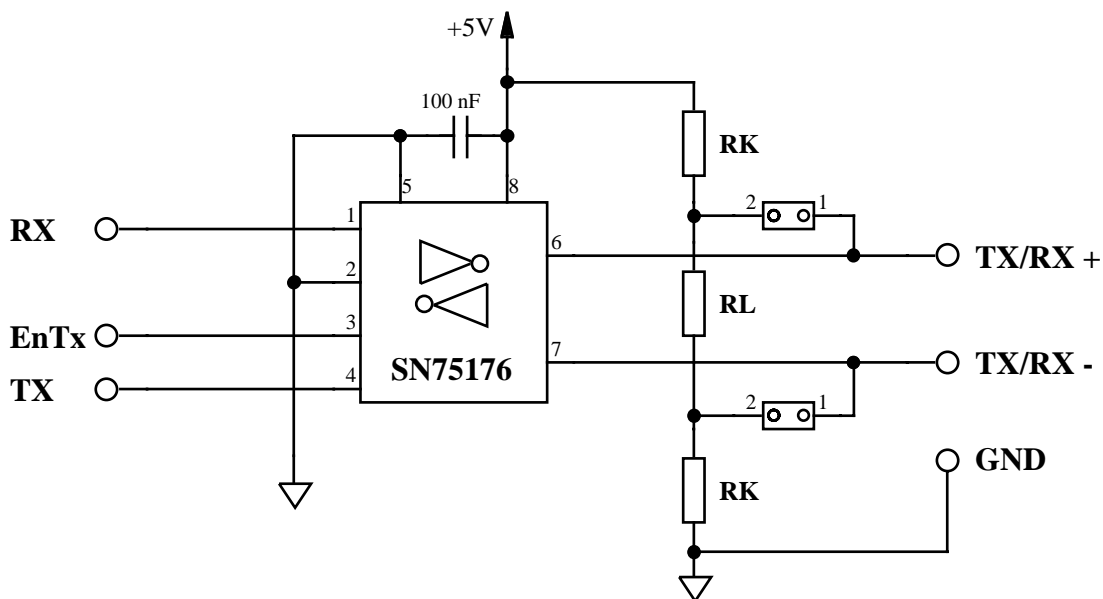



FIGURA 3: CIRCUITERIA DI INTERFACCIAMENO IN RS485 BASATA SUL SN75176

Le resistenze **RL** e **RK**, compongono la circuiteria di **terminazione** della linea; questa, che può venire inserita, chiudendo i due jumper appositi, deve essere presente solo nei dispositivi posti rispettivamente all'inizio (l'unità **master**) ed alla fine (il dipositivo **slave n-esimo**) della rete di **n** elementi collegati in **Daisy-Chain**.
Per determinare i valori di tali componenti, bisogna considerare che la linea di comunicazione deve essere caricata con una resistenza di **60Ω**, e che le resistenze di forzatura verso Vcc e GND devono avere un valore compreso tra **5.6 KΩ** e **6.8 KΩ**.

Per questi motivi i valori di **RL** e **RK** sono i seguenti:

*Resistenza **RL**:* Tale componente deve essere presente sia sul master che sullo slave n-esemo (dispositi agli estremi della linea), quindi deve essere: **RL = 120 Ω**.

*Resistenza **RK**:* Questi componenti, per comodità di montaggio, possono essere omessi dal master o dallo slave n-esimo; in questo caso si avrà che: **RK = 5.6 KΩ÷6.8 KΩ**.
Se invece, tali resistenze sono presenti su entrambi i dispositivi, dovrà essere: **RK = 2.7 KΩ÷3.3 KΩ**.

Nella figura seguente é riportato uno schema esemplificativo, che illustra una catena Daisy-Chain, con una un'unità Master, ed **n** unità Slave.

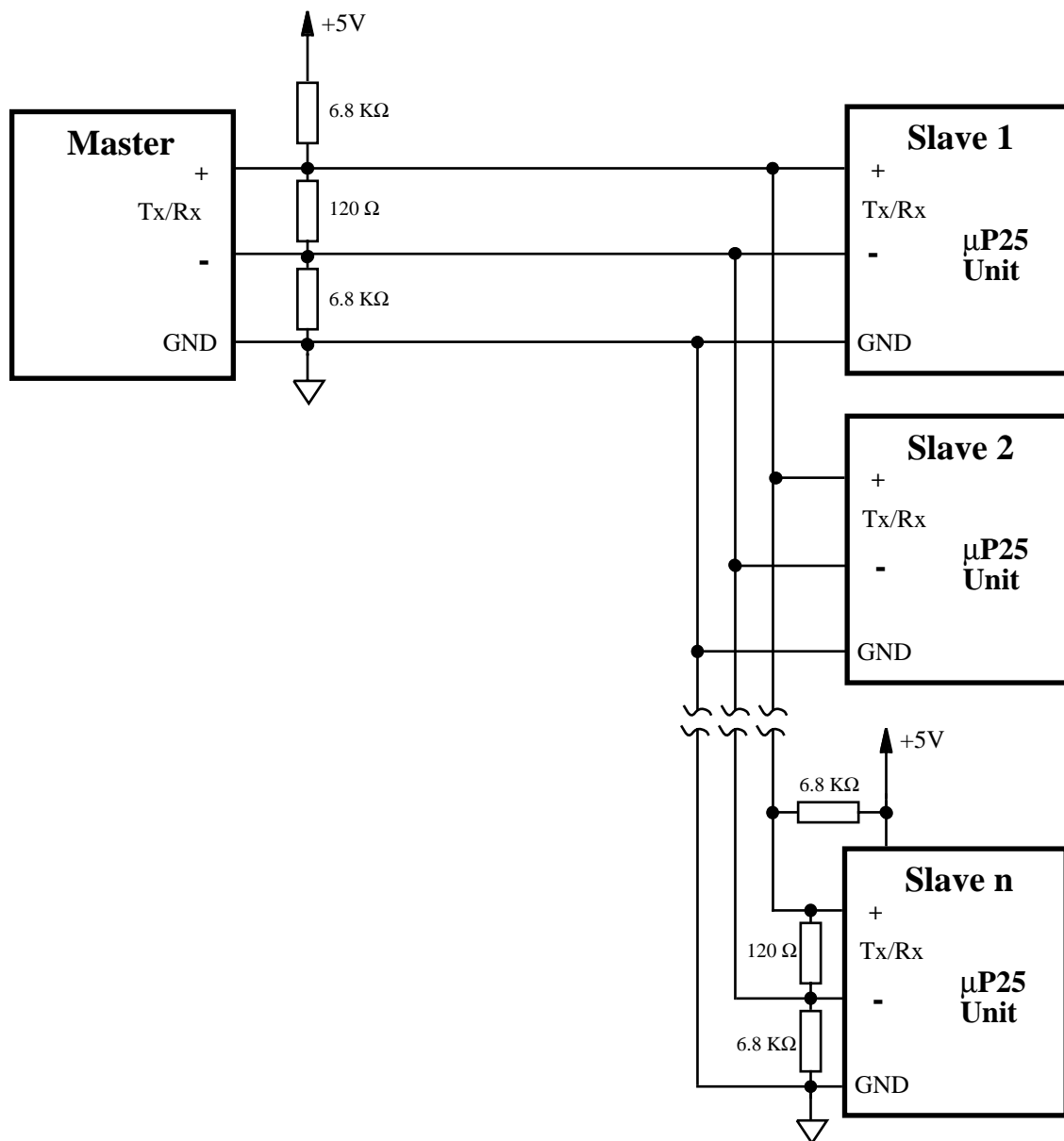



FIGURA 4: ESEMPIO DI COLLEGAMENTO IN RETE PER COMUNICAZIONE MASTER-SLAVE

CARATTERISTICHE GENERALI

Risorse di bordo:
- Modalità di SETUP locale per la completa configurazione del sistema.
- Fino a **25 linee** gestibili per operazioni di I/O, con possibilità di generare segnali quali onda quadra, settaggio o resettaggio temporizzato di un pin,ecc., il tutto gestito ad alto livello.
- Supporta varie **EEPROM** seriali fino a **2KBytes**, per il salvataggio di Setup e messaggi.
- Fino a **203 stringhe** di 10 caratteri memorizzabili nella EEPROM.
- **Counter a 16 bit** gestito ad alto livello, a disposizione utente.
- Gestione ad alto livello dell'**A/D converter a 12 bit TLC 2543**.
- Gestione ad alto livello del **D/A converter a 16 bit AD 420**.
- Gestione ad alto livello del **LED display driver M5480**.

CPU utilizzata: 87c51.

Protocollo di comunic.: Protocollo di comunicazione per modalità punto-punto o Master-Slave.
BAUD RATE: 1200, 2400, 4800, 9600, 19200 o 38400 Baud.
1 Stop Bit.
No Parity.
8 Bit (modalità punto-punto), 9 Bit (modalità Master-Slave).
Fino a 128 dispositivi **μP25** collegabili in rete in modalità Master-Slave.

CARATTERISTICHE FISICHE

Contenitore: PLCC 44 pin.

Range di temperatura: Da 0 a 70 gradi centigradi.

Umidità relativa: 20% fino a 90% (senza condensa).

CARATTERISTICHE ELETTRICHE

Tensione di alimentaz.: +5 Vdc.

Frequenza del Quarzo: 14.7456 MHz.

Durata min. del Reset: 5 μsec

DESCRIZIONE SOFTWARE DEL µP25

Il dispositivo **µP25**, come già detto, é un dispositivo di I/O gestibile tramite una serie di comandi sulla linea seriale; per questo motivo tutto quanto ricevuto viene interpretato ed eseguito, ed eventuali codici di risposta vengono restituiti all'unità master di controllo.
Il firmware del **µP25** prevede inoltre, una modalità di funzionamento Setup, che permette di configurare, nel modo desiderato, tutte le varie sezioni del dispositivo.
Di seguito, oltre alla descrizione delle funzionalità del **µP25**, viene riportato un elenco completo di quelle che sono le sequenze di comando e le combinazioni riconosciute, da utilizzare per usufruire di tutte le principali caratteristiche del dispositivo. Per ogni codice, o sequenza di codici, viene riportata una doppia descrizione: quella mnemonica, tramite caratteri ASCII e quella numerica espressa in forma decimale ed esadecimale.

SELEZIONE DELLA MODALITA' DI FUNZIONAMENTO

Il **µP25** é in grado di funzionare con due modalità distinte, in particolare può essere attivato il **modo SETUP**, oppure il **modo RUN**. La selezione dell'uno o dell'altro funzionamento,avviene durante la fase di Power-ON, verificando lo stato logico in cui si trova il **/SETUP** (pin 15) del **µP25**; in particolare:

/SETUP (pin 15) = Livello logico **0** -> *Modo SETUP*
/SETUP (pin 15) = Livello logico **1** -> *Modo RUN*

Di seguito viene riportata una descrizione completa, dei comandi disponibili, nelle due modalità di funzionamento del **µP25**.

MODALITA' DI FUNZIONAMENTO SETUP

Tramite il modo **SETUP** é possibile definire tutti i vari parametri d' inizializzazione, relativi al settaggio dei Port, tipo di comunicazione, Baud Rate, Nome del dispositivo, tipo di EEPROM montata, ecc.
Tali dati, che il **µP25** andrà a memorizzare all'interno della EEPROM, permettono di definire la configurazione con cui lo stesso lavorerà nel modo RUN.
In questa modalità di funzionamento, tutti i Port vengono settati in INPUT, inoltre il protocollo di comunicazione é fisso ed é il seguente:

9600 Baud, 8 Bit , 1 Stop, NO Parity, comunicazione Punto-Punto

In quesa fase non é quindi possibile utilizzare la comunicazione **Master-Slave**.

COMANDI RELATIVI ALLA MODALITA' DI FUNZIONAMENTO SETUP

Sono di seguito riportati i comandi che il **μP25** riconosce quando il esegue la modalità **SETUP**:

SETTAGGIO DEL TIPO DI EEPROM

Sequenza di Input:

Codice:	***69 <codice EEPROM>***	***(45 <codice EEPROM> Hex)***
Mnemonico:	***E ASCII(codice EEPROM)***	

Viene settato il tipo di EEPROM installata, secondo quando indicato in "codice EEPROM".
I valori che tale byte può assumere sono i seguenti:

1	*EEPROM del tipo 24C02*
2	*EPPROM del tipo 24C04*
4	*EEPROM del tipo 24C16*

Esempio:
Se si vuole selezionare una EEPROM 24C04, sarà necessario inviare la seguente sequenza: **69 2**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il "codice EEPROM" é corretto o meno, tale risposta può essere:

6	*(ACK)*	*Codice EEPROM valido*
21 - 15 Hex	*(NACK)*	*Codice EEPROM non valido: Comando ignorato*

Se il "codice EEPROM" é valido, questo verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato in seguito.

SETTAGGIO DEL BAUD RATE DI COMUNICAZIONE

Sequenza di Input:

Codice:	***66 <codice BAUD>***	***(42 <codice BAUD> Hex)***
Mnemonico:	***B ASCII(codice BAUD)***	

Viene settato il Baud Rate di comunicazione, relativo alla modalità RUN, secondo quando indicato in "codice BAUD". I valori che tale byte può assumere sono i seguenti:

0	*Comunicazione a 1200 Baud*
1	*Comunicazione a 2400 Baud*
2	*Comunicazione a 4800 Baud*
3	*Comunicazione a 9600 Baud*
4	*Comunicazione a 19200 Baud*
5	*Comunicazione a 38400 Baud*

Esempio:
Se si vuole selezionare una comunicazione a 19200 Baud, sarà necessario inviare la seguente sequenza: **66 4**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il “codice BAUD” é corretto o meno, tale risposta può essere:

6	*(ACK)*	*Codice BAUD valido*
21 - 15 Hex	*(NACK)*	*Codice BAUD non valido: Comando ignorato*

Se il “codice BAUD” é valido, questo verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato in seguito.

SETTAGGIO DEL TIPO DI COMUNICAZIONE

Sequenza di Input:

Codice: ***67 <codice Comunicazione> (43 <codice Comunicazione> Hex)***
Mnemonico: ***C ASCII(codice Comunicazione)***

Viene settata la modalità di comunicazione, durante l'esecuzione del modo RUN, secondo quanto inidcato in “codice Comunicazione”. I valori che tale byte può assumere sono i seguenti:

0	*Comunicazione PUNTO-PUNTO*
1	*Comunicazione MASTER-SLAVE a 9 Bit*

Esempio:
Se si vuole selezionare la comunicazione Master-Slave a 9 Bit, sarà necessario inviare la seguente sequenza: **67 1**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il “codice Comunicazione” é corretto, tale risposta può essere:

6	*(ACK)*	*Codice Comunicazione valido*
21 - 15 Hex	*(NACK)*	*Codice Comunicazione non valido: Comando ignorato*

Se il “codice Comunicazione” é valido, questo verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato di seguito.

SETTAGGIO NOME PER COMUNICAZIONE MASTER-SLAVE

Sequenza di Input:

Codice: ***78 <Nome> (4E <Nome> Hex)***
Mnemonico: ***N ASCII(Nome)***

Viene settato il“ Nome” al quale il **µP25** deve rispondere se, nel modo RUN, é settato per comunicare in modalità Master-Slave.
I valori che tale byte può assumere sono tutti quelli compresi nel campo: ***128..255 (80..FF Hex)***.

Esempio:
Se si vuole settare il nome con il valore 170, sarà necessario inviare la seguente sequenza: **78 170**.

Codici di risposta:
Dopo aver inviato tale comando,il **µP25**, restituisce un byte che indica se il “Nome” é corretto, tale risposta può essere:

6	*(ACK)*	*Nome valido*
21 - 15 Hex	*(NACK)*	*Nome non valido: Comando ignorato*

Se il “Nome” é valido, questo verrà memorizzato nella RAM del **µP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato di seguito.

SETTAGGIO DEI PORT

Sequenza di Input:

Codice: ***80 <Port> <byte settaggio> (50 <Port> <byte settaggio> Hex)***
Mnemonico: ***P ASCII(Port) ASCII(byte settaggio)***

Mediante questo comando, viene memorizzato quello che dovrà essere il settaggio del Port indicato in “Port”, quando il **µP25** si trova nel modo RUN. Tale byte può avere i seguenti valori: **0, 1, 2**.
Il port in questione sarà inizializzato secondo quanto indicato nel codice “byte settaggio”.
Il significato di tale byte é il seguente:

(bit 7) Px.7 Px.6 Px.5 Px.4 Px.3 Px.2 Px.1 Px.0 (bit 0)

Dove ogni bit di tale byte, indica il settaggio che deve avere il relativo bit del Port indicato; in particolare:

Px.?	*0*	*Il bit in questione del port indicato é settato in INPUT*
	1	*Il bit in questione del port indicato é settato in OUTPUT*

Se si desidera attivare la gestione di A/D, D/A o M5480, i bit del Port 2 utilizzati per comandare la relativa periferica, possono essere settati indifferentemente in Input o in Output.

Esempio:
Se si vuole settare il Port 1 con i bit 0-3 in Input ed i bit 4-7 in Output, sarà necessario inviare la seguente sequenza: **80 1 240**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il comando inviato é corretto, tale risposta può essere:

6	*(ACK)*	*Settagio Port valido*
21 - 15 Hex	*(NACK)*	*Settaggio Port non valido: Comando ignorato*

Se il comando é valido, il settaggio del Port in questione, verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato di seguito.

SETTAGGIO DEL PIN INT0

Sequenza di Input:

Codice: ***73 <codice INT0> (49 <codice INT0> Hex)***
Mnemonico: ***I ASCII(codice INT0)***

Tramite il valore di "codice INT0", viene indicato quello che dovrà essere il settaggio del pin INT0 (pin **14**) del **μP25**, quando questo si trova nel modo RUN.
I valori che tale byte può assumere sono i seguenti:

0	*Pin INT0 settato come linea di INPUT*
1	*Pin INT0 settato come linea di OUTPUT*
2	*Pin INT0 settato come Trigger per il Counter a 16 Bit*

Esempio:
Se si vuole settare il pin INT0 del **μP25**, come Trigger del counter a 16 bit, sarà necessario inviare la seguente sequenza: **73 2**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il "codice INT0" é corretto, tale risposta può essere:

6	*(ACK)*	*Codice INT0 valido*
21 - 15 Hex	*(NACK)*	*Codice INT0 non valido: Comando ignorato*

Se il "codice INT0" é valido, questo verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato di seguito.

ATTIVAZIONE DELLA GESTIONE DI A/D, D/A E M5480

Sequenza di Input:

Codice: ***70 <codice A.D.M.> (46 <codice A.D.M.> Hex)***
Mnemonico: ***F ASCII(codice A.D.M.)***

Permette di indicare al **μP25** se, nel modo RUN, dovrà attivare o meno la gestione, ad alto livello, dell'A/D converter **TLC2543**, del D/A converter **AD420** o del LED display driver **M5480**.
Tali indicazioni sono contenute in "codice A.D.M.", il cui significato é il seguente:

Bit 0 *Gestione A/D converter TLC 2543* ***On/Off***
Bit 1 *Gestione D/A converter AD 420* ***On/Off***
Bit 2 *Gestione LED display driver M5480* ***On/Off***
Bit 3÷7 *Non utilizzati:* ***Devono essere posti a 0***

I Bit 0, 1 e 2 di tale byte indicano se deve essere attivata, o meno, la gestione del dispositivo relativo, in particolare:

Bit x *0* *Gestione dispositivo DISATTIVATA*
1 *Gestione dispositivo ATTIVATA*

Esempio:
Se si desidera settare il **μP25** per gestire A/D converter e M5480, sarà necessario inviare la seguente sequenza: **70 5**.

Codici di risposta:
Dopo aver inviato tale comando,il **μP25**, restituisce un byte che indica se il "codice A.D.M." é corretto, tale risposta può essere:

6 *(ACK)* *Codice A.D.M. valido*
21 - 15 Hex *(NACK)* *Codice A.D.M. non valido: Comando ignorato*

Se il "codice A.D.M." é valido, questo verrà memorizzato nella RAM del **μP25**, e sarà scritto in EEPROM solo a seguito del comando di **Salvataggio Settaggio (S)**, illustrato di seguito.

LETTURA DELLA CONFIGURAZIONE ATTUALE

Sequenza di Input:

Codice:	***76***	***(4C Hex)***
Mnemonico:	***L***	

Questo comando permette di acquisire i codici relativi alla settaggio corrente del **µP25**.

Codici di risposta:
Vengono restituiti 9 byte che contengono i seguenti codici:

Byte 1	Codice EEPROM
Byte 2	Codice BAUD
Byte 3	Codice Comunicazione
Byte 4	NOME **µP25** per comunicazione Master-Slave
Byte 5	Settaggio PORT 0
Byte 6	Settaggio PORT 1
Byte 7	Settaggio PORT 2
Byte 8	Codice INT0
Byte 9	Codice A.D.M.

Il significato di questi 9 byte restituiti dal **µP25** é quello riportato nei paragrafi precedenti.

SALVATAGGIO SETTAGGIO IN EEPROM

Sequenza di Input:

Codice:	***83***	***(53 Hex)***
Mnemonico:	***S***	

Il **µP25** memorizza, in EEPROM, tutti i dati relativi al settaggio, attualmente contenuti nella RAM di bordo , quindi restituisce l'esito dell'operazione e si pone il un loop di attesa; <u>a questo punto é possibile solo togliere alimentazione dal dispositivo</u>.
Se un parametro non é stato variato, in EEPROM rimarà il valore che questo aveva precedentemente.

Codici di risposta:
Dopo aver salvato i parametri ,il **µP25**, restituisce un byte che indica se tale operazione ha avuto successo o meno; queta risposta può essere:

6	*(ACK)*	*Salvataggio dei parametri in EEPROM senza errori*
21 - 15 Hex	*(NACK)*	*Errore durante il salvataggio dei parametri in EEPROM*

TABELLA RIASSUNTIVA DEI COMANDI DEL MODO SETUP

Viene riportata di seguito la tabella riassuntiva dei comandi disponibili nel modo SETUP.

COMANDO	CODICE	CODICE HEX	MNEMONICO
Settaggio EEPROM	69 cod.EE	45 cod.EE	E ASCII(cod.EE)
Settaggio BAUD RATE	66 cod.Baud	42 cod.Baud	B ASCII(cod.Baud)
Settaggio tipo di COMUNICAZIONE	67 cod.Com	43 cod.Com	C ASCII(cod.Com)
Settaggio NOME	78 nome	4E nome	N ASCII(nome)
Settaggio PORT	80 port byte	50 port byte	P ASCII(port) ASCII(byte)
Settaggio pin INT0	73 cod.Int0	49 cod.Int0	I ASCII(cod.Int0)
Settaggio A/D, D/A e M5480	70 cod.	4D cod.	M ASCII(cod.)
Lettura configurazione attuale	76	4C	L
Salvataggio settaggio	83	53	S

FIGURA 5: TABELLA DEI COMANDI RELATIVI AL MODO SETUP

MODALITA' DI FUNZIONAMENTO RUN

Quando si accede al modo **RUN**, vengono verificati i parametri memorizzati nell'EEPROM, relativi all'inizializzazione dei Port , al protocollo di comunicazione, ecc.
Se tali dati sono validi, il **µP25** setta a livello logico **0** la linea **/SETUP** (pin **15**), quindi procede con le varie inizializzazioni e quindi si mette nella fase di attesa comando.
In caso contrario, vengono settati tutti i Port in Input, viene settato a livello logico **1** la linea **/SETUP** (pin **15**), quindi il **µP25** entra in un loop di attesa; a questo punto é possibile solo togliere alimentazione dal dispositivo.

Il protocollo di comunicazione del modo RUN é settabile in termini di Baud Rate e Tipo di Comunicazione (mediante la modalità SETUP), mentre il formato dei dati é in funzione della modalità selezionata, in particolare:

Comunicazione Punto-Punto:	***8 bit, 1 Stop, NO Parity***
Master-Slave:	***9 bit, 1 Stop, NO Parity***

I comandi riconosciuti nella modalità di funzionamento **RUN**, sono descritti di seguito.

COMANDO DI RESET GENERALE

Sequenza di Input:

Codice:	***65 97***	***(41 61 Hex)***
Mnemonico:	***A a***	

A seguito di questo comando, il **µP25** ripristina la condizione di inizializzazione che si verifica all'atto del Power-ON; in paricolare:

Port 0,1, e 2	Vengono resettati e posti nella condizione iniziale, quindi con tutti i pin di Output settati allo stato logico 0. Eventuali temporizzazioni su bit del Port 0 vengono interrotte.
Pin INT0	Se settato in Output viene posto allo stato logioco 0.
Counter a 16 Bit	Viene inizializzato con il valore 0.
D/A Conveter	Se la gestione di tale dispositivo é attiva, viene inizializzato con la combinazione 0.
M5480	Se la gestione di questo dispositivo é attiva, vengono resettate tutte le sue uscite.

COMANDI RELATIVI ALLA GESTIONE DEI PORT

Sono riportati di seguito i comandi per la gestione, lettura e scrittura, dei Port 0, 1 e 2 del **μP25**.

SCRITTURA DI UN PORT

Sequenza di Input:

Codice: ***87 <Port> <nibble L Dato> <nibble H Dato>***
(57 <Port> <nibble L Dato> <nibble H Dato> Hex)
Mnemonico: ***W ASCII(Port) ASCII(nibble L Dato) ASCII(nibble H Dato)***

Il dato formato dai due nibble L e H viene scritto sul Port indicato in “Port”. Tale byte può avere i seguenti valori: **0, 1, 2**. Il byte “Dato” deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Px.3	Px.2	Px.1	Px.0	(LSB)
Nibble H:		0	0	0	0	Px.7	Px.6	Px.5	Px.4	

Dove i bit **Px.?** di tali bytes, indicano gli stati logici, 0 o 1, che devono assumere i relativi bit del Port indicato.
Nel port in questione saranno posti allo stato indicato, solo i bit inizializzati in Output; se tutto il port é settato in Input il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.
Se é attivata la gestione ad alto livello di A/D, D/A o M5480, il comando di scrittura relativo al **Port 2**, andrà a comandare solo i bit che non sono utilizzati per tali funzioni.

Esempio:
Sesi vuole scrivere il dato 90 (5A Hex) sul Port 0, sarà necessario inviare la seguente sequenza: **87 0 10 5**.

Note:
Se il port in questione non é completamente settato in Output, ma vi sono alcuni bit inizializzati in Input, in corrispondenza di un comando di questo tipo la scrittura di tutti bit di uscita, con lo stato logico indicato, non avverrà contemporaneamente bensì in sequenza.
In questo caso, il ritardo tra la scrittura di un bit ed il successivo, sarà di circa **50 μsec**.

LETTURA DI UN PORT

Sequenza di Input:

Codice: ***82 <Port>*** ***(52 <Port> Hex)***
Mnemonico: ***R ASCII(Port)***

Viene restituito in seriale il dato letto dal Port indicato in “Port”. Tale byte può avere i seguenti valori: **0, 1, 2**.
Nel port in questione saranno acquisiti solo gli stati logici dei bit inizializzati in Input, mentre saranno posti a 0 i bit **Px.?**, relativi ai pin settati in Output; se tutto il port é settato in Output si avrà quindi come risposta il dato 0.

Codici di Risposta:
Il dato acquisito dal Port viene restituito sotto forma di nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Px.3	Px.2	Px.1	Px.0	(LSB)
Nibble H:		0	0	0	0	Px.7	Px.6	Px.5	Px.4	

Dove i bit **Px.?** di tali bytes, indicano gli stati logici, 0 o 1, a cui si trovano i relativi bit del Port indicato.

Note:
Nel Port in questione saranno acquisiti solo gli stati logici dei bit configurati in Input, mentre saranno posti a **0** tutti quelli settati in Output, oppure utilizzati per la gestione ad alto livello di A/D, D/A o M5480. Se invece, la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole leggere il Port 1, sul quale é presente il dato 90 (5A Hex) sarà necessario inviare la seguente sequenza: **82 1**, e come risposta si avranno i seguenti bytes: **10 5**.

LETTURA DELLA CONFIGURAZIONE DEI PORT

Sequenza di Input:

Codice: ***70*** ***(46 Hex)***
Mnemonico: ***F***

Vengono restituiti i byte di configurazione dei Port 0, 1 e 2, del **µP25**, precedentemente determinati tramite il modo SETUP.

Codici di risposta:
I 3 byte di configurazione dei Port 0, 1 e 2 sono restituiti a nibble, in particolare vengono trasmessi dal **µP25**, sei codici con il seguente significato:

Port 0	(nibble L):	(MSB)	0	0	0	0	P0.3	P0.2	P0.1	P0.0	(LSB)
	(nibble H) :		0	0	0	0	P0.7	P0.6	P0.5	P0.4	
Port 1	(nibble L):		0	0	0	0	P1.3	P1.2	P1.1	P1.0	
	(nibble H) :		0	0	0	0	P1.7	P1.6	P1.5	P1.4	
Port 2	(nibble L):		0	0	0	0	P2.3	P2.2	P2.1	P2.0	
	(nibble H) :		0	0	0	0	P2.7	P2.6	P2.5	P2.4	

Dove ogni bit di tali byte, indica la configurazione attuale del relativo Pin del Port; in particolare:

Px.?	*0*	*Il bit in questione del port indicato é configurato in INPUT*
	1	*Il bit in questione del port indicato é configurato in OUTPUT*

COMANDI RELATIVI ALLA GESTIONE DEI SINGOLI BIT DEI PORT

Sono riportati di seguito i comandi per la gestione dei singoli Pin dei Port 0, 1 e 2 del μ**P25**.

SET DEL BIT DI UN PORT

Sequenza di Input:

Codice: ***83 <Port> <Bit> (53 <Port> <Bit> Hex)***
Mnemonico: ***S ASCII(Port) ASCII(Bit)***

Viene posto allo stato logico **1**, il Pin indicato in “Bit” relativo al Port indicato in “Port”.
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0**, **1**, **2** e **3**.

Se il Pin **INT0** é configurato in Output e lo desidera settare, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Note:
Se sul Pin in questione desiderato sono in corso delle temporizzazioni, queste vengono interrotte.
Invece, se tale Bit é settato in Input, oppure é utilizzato per la gestione ad alto livello di A/D, D/A o M5480, il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole settare il Bit 4 del Port 0, sarà necessario inviare la seguente sequenza: **83 0 4**.

CLEAR DEL BIT DI UN PORT

Sequenza di Input:

Codice: ***67 <Port> <Bit> (43 <Port> <Bit> Hex)***
Mnemonico: ***C ASCII(Port) ASCII(Bit)***

Viene posto allo stato logico 0, il Pin indicato in “Bit” relativo al Port indicato in “Port”.
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0**, **1**, **2** e **3**.

Se il Pin **INT0** é configurato in Output e lo desidera resettare, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Note:
Se sul Pin in questione desiderato sono in corso delle temporizzazioni, queste vengono interrotte.
Invece, se tale Bit é settato in Input, oppure é utilizzato per la gestione ad alto livello di A/D, D/A o M5480, il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole resettare il Bit 7 del Port 2, sarà necessario inviare la seguente sequenza: **67 2 7**.

SET TEMPORIZZATO DEL BIT DI UN PORT

Sequenza di Input:

Codice:	***115 <Port> <Bit> <nibble L Cicli> <nibble H Cicli>***
	(73 <Port> <Bit> <nibble L Cicli> <nibble H Cicli> Hex)
Mnemonico:	***s ASCII(Port) ASCII(Bit) ASCII(nib. L Cicli) ASCII(nib. H Cicli)***

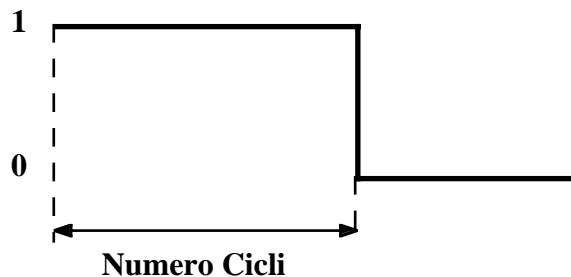



FIGURA 6: COMANDO DI SET TEMPORIZZATO PORT.BIT

Viene posto allo stato logico 1, per un determinato tempo, il Pin indicato in “Bit” relativo al Port indicato in “Port”.
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0** e **3**; sui rimanti Port non si può eseguire nessun tipo di comando di Output temporizzato.

Se si desidera applicare questo comando sul il Pin **INT0**, configurato in Output, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Il Pin in questione del Port indicato rimane allo stato logico 1 per un tempo determinato dal byte “Cicli”, quindi ritorna allo stato logico 0.
Tale byte, che deve essere inviato a nibble secondo il seguente formato:

Nibble L:	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
Nibble H:		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Il byte “Cicli” deve essere compreso nel range **1..255**, ed il tempo relativo si può facilmente determinare, sapendo che 1 ciclo corrisponde a **10 msec**.

Note:
Se il Bit in questione del Port indicato é settato in Input il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole settare il pin INT0, per un tempo di 500 msec, corrispondente a 50 cicli, sarà necessario inviare la seguente sequenza: **115 3 0 2 3**.

CLEAR TEMPORIZZATO DEL BIT DI UN PORT

Sequenza di Input:

Codice: ***99 <Port> <Bit> <nibble L Cicli> <nibble H Cicli>***
(63 <Port> <Bit> <nibble L Cicli> <nibble H Cicli> Hex)

Mnemonico: ***c ASCII(Port) ASCII(Bit) ASCII(nib. L Cicli) ASCII(nib. H Cicli)***

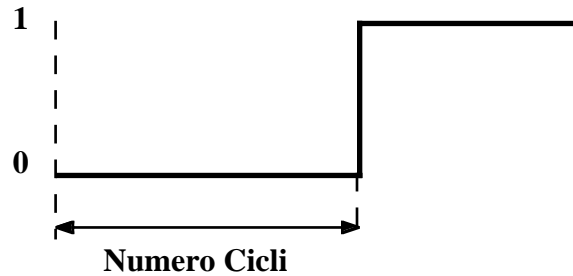



FIGURA 7: COMANDO DI CLEAR TEMPORIZZATO PORT.BIT

Viene posto allo stato logico 0, per un determinato tempo, il Pin indicato in “Bit” relativo al Port indicato in “Port”.
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0** e **3**; sui rimanti Port non si può eseguire nessun tipo di comando di Output temporizzato.

Se si desidera applicare questo comando sul il Pin **INT0**, configurato in Output, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Il Pin in questione del Port indicato rimane allo stato logico 0 per un tempo determinato dal byte “Cicli”, quindi ritorna allo stato logico 1.
Tale byte, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **1..255**, ed il tempo relativo si può facilmente determinare, sapendo che 1 ciclo corrisponde a **10 msec**.

Note:
Se il Bit in questione del Port indicato é settato in Input il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole settare il Bit 7 del Port 0, per un tempo di 1 sec, corrispondente a 100 cicli, sarà necessario inviare la seguente sequenza: **99 0 7 4 6**.

ONDA QUADRA SU UN BIT DI UN PORT

Sequenza d i Input:

Codice: ***80 <Port> <Bit> <nibble L Cicli> <nibble H Cicli>***
(50 <Port> <Bit> <nibble L Cicli> <nibble H Cicli> Hex)

Mnemonico: ***P ASCII(Port) ASCII(Bit) ASCII(nib. L Cicli) ASCII(nib. H Cicli)***

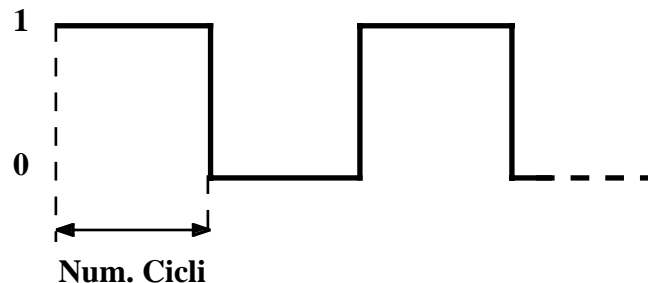



FIGURA 8: COMANDO DI ONDA QUADRA SU PORT.BIT

Viene generata un'onda quadra con Duty Cicle del 50%, sul Pin indicato in "Bit" relativo al Port indicato in "Port".
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0** e **3**; sui rimanti Port non si può eseguire nessun tipo di comando di Output temporizzato.

Se si desidera applicare questo comando sul il Pin **INT0**, configurato in Output, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Il periodo del segnale generato, viene determinato dal valore del byte "Cicli", che esprime il tempo di durata di una semi-onda.
Tale byte, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **1..255**, ed il tempo relativo si può facilmente determinare, sapendo che 1 ciclo corrisponde a **10 msec**.

Note:
Se il Bit in questione del Port indicato é settato in Input il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole generare un'onda quadra sul Bit 2 del Port 0, con periodo di 200 msec, corrispondente a 10 cicli, sarà necessario inviare la seguente sequenza: **80 0 2 10 0**.

ONDA QUADRA TEMPORIZZATA SU UN BIT DI UN PORT

Sequenza di Input:

Codice: ***112 <Port> <Bit>***
<nib. L Cicli> <nib. H Cicli> <nib. L Stati> <nib. H Stati>
(70 <Port> <Bit>
<nib. L Cicli> <nib. H Cicli> <nib. L Stati> <nib. H Stati> Hex)

Mnemonico: ***p ASCII(Port) ASCII(Bit) ASCII(nib. L Cicli) ASCII(nib. H Cicli) ASCII(nib. L Stati) ASCII(nib. H Stati)***

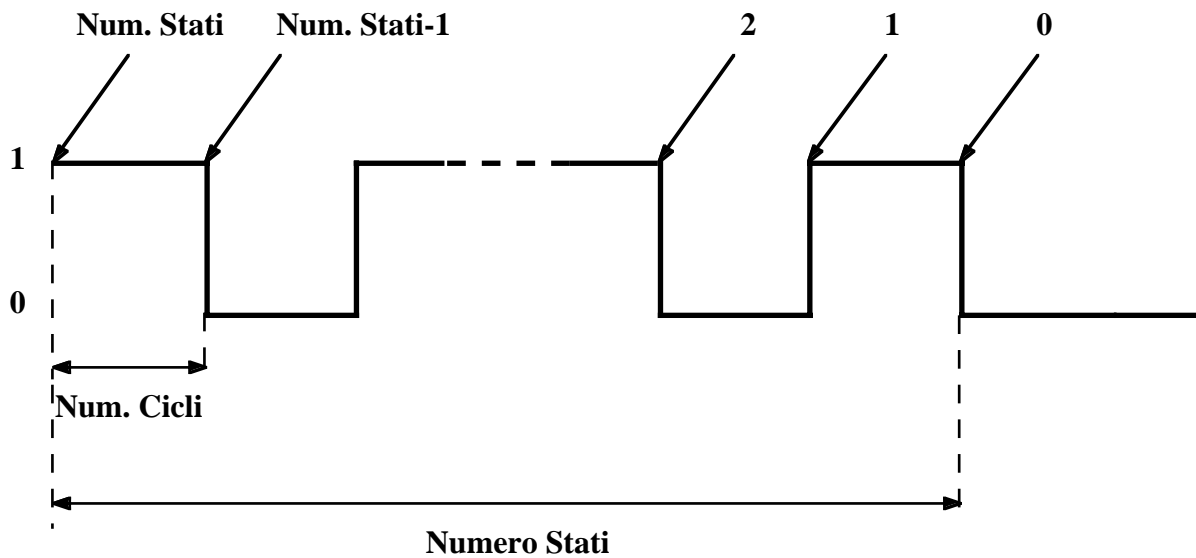



FIGURA 9: COMANDO DI ONDA QUADRA TEMPORIZZATA SU PORT.BIT

Viene generata un'onda quadra con Duty Cicle del 50%, per un determinato tempo, sul Pin indicato in "Bit" relativo al Port indicato in "Port".
Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0** e **3**; sui rimanti Port non si può eseguire nessun tipo di comando di Output temporizzato.

Se si desidera applicare questo comando sul il Pin **INT0**, configurato in Output, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Il periodo del segnale generato, viene determinato dal valore del byte "Cicli", che esprime il tempo di durata di una semi-onda.
Tale byte, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **1..255**, ed il tempo relativo si può facilmente determinare, sapendo che 1 ciclo corrisponde a **10 msec**.

Il tempo di permanenza di tale segnale sul Pin in questione é determinato dal byte "Stati", che deve essere inviato anch'esso a nibble secondo il formato indicato in precedenza, e deve essere compreso nel range **1..255**.
Mediante questo byte é infatti possibile indicare il numero di cambiamenti di stato che devono avenire sul Pin indicato, prima che quest'ultimo ritorni definitavente allo stato logico 0; come si può notare dalla Figura, il numero di cambiamenti di stato é: "Stati"+1.

Note:
Se il Bit in questione del Port indicato é settato in Input il comando viene ignorato; questo avviene anche nel caso che la sequenza inviata contenga dei dati non validi.

Esempio:
Se si vuole generare un'onda quadra sul pin INT0, con periodo di 200 msec, corrispondente a 10 cicli, e con 10 cambiamenti di stato, sarà necessario inviare la seguente sequenza:
112 3 0 10 0 9 0.

LETTURA DEL BIT DI UN PORT

Sequenza di Input:

Codice: ***114 <Port> <Bit> (72 <Port> <Bit> Hex)***
Mnemonico: ***r ASCII(Port) ASCII(Bit)***

Viene restituito lo stato logico, 0 o 1, del Pin indicato in "Bit" relativo al Port indicato in "Port". Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0**, **1**, **2** e **3**.

Se il Pin **INT0** é configurato in Input e si desidera acquisirne lo stato, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Codici di Riposta:
Viene restituito lo stato del Pin richiesto, in particolare:

0 *Il Pin si trova allo stato logico 0*
1 *Il Pin si trova allo stato logico 1*

Note:
Lo stato logico del pin in questione sarà acquisito solo nel caso che questo sia configurato in Input; se tale Bit é settato in Output, oppure é utilizzato per la gestione ad alto livello di A/D, D/A o M5480, verrà restituito in codice **0**
Invece, se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole leggere lo stato del Bit 7 del Port 2, il quale si trova allo stato logico 1, sarà necessario inviare la seguente sequenza: **114 2 7**, e sarà restituito il valore **1**.

LETTURA CON DEBAUNCING DEL BIT DI UN PORT

Sequenza di Input:

Codice: ***68 <Port> <Bit> <nibble L Cicli> <nibble H Cicli>***
(44 <Port> <Bit> <nibble L Cicli> <nibble H Cicli> Hex)
Mnemonico: ***D ASCII(Port) ASCII(Bit) ASCII(nib. L Cicli) ASCII(nib. H Cicli)***

Viene restituito lo stato logico, acquisito in continuazione durante un determinato tempo di debauncing , del Pin indicato in "Bit" relativo al Port indicato in "Port".

Questi bytes possono avere i seguenti valori: **0..7**, per quel che riguarda il numero del Bit, mentre i valori validi per il numero del Port sono **0**, **1**, **2** e **3**.

Se il Pin **INT0** é configurato in Input e si desidera acquisirne lo stato, sarà necessario indicare il **Port 3** ed il **Bit 0** (tutti gli altri bit del Port 3 sono considerati codici errati).

Il tempo di debauncing é determinato dal valore del byte "Cicli".
Tale byte, che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **1..255**, ed il tempo relativo si può facilmente determinare, sapendo che 1 ciclo corrisponde a **10 msec**.

Codici di Riposta:
Il dato che viene restituito, può assumere i seguenti valori:

0	*Il Pin é sempre stato allo stato logico 0 durante tutto il tempo di debauncing*
1	*Il Pin é sempre stato allo stato logico 1 durante tutto il tempo di debauncing*
7	*Lo stato logico del pin é variato prima che scadesse il tempo di debauncing*

Note:
Lo stato logico del pin in questione sarà acquisito solo nel caso che questo sia configurato in Input; se tale Bit é settato in Output, oppure é utilizzato per la gestione ad alto livello di A/D, D/A o M5480, verrà restituito in codice **0**
Invece, se la sequenza inviata contiene dei dati non validi, il comando viene ignorato.

Esempio:
Se si vuole leggere lo stato del Bit 3 del Port 1, il quale si trova allo stato logico 0, con un tempo di debauncing di 50 msec, corrispondente a 5 cicli, sarà necessario inviare la seguente sequenza:
114 1 3 5 0.

LETTURA DELLA CONFIGURAZIONE DEL PIN INT0

Sequenza di Input:

Codice:	***102***	***(66 Hex)***
Mnemonico:	***f***	

Viene restituito il codice di configurazione del Pin INT0, del **µP25**, precedentemento determinato tramite il modo SETUP.

Codici di risposta:
Il byte di configurazione del pin INT0, che viene restituito dal **µP25** può avere i seguenti valori:

0	*Pin INT0 settato come linea di INPUT*
1	*Pin INT0 settato come linea di OUTPUT*
2	*Pin INT0 settato come Trigger per il Counter a 16 Bit*

COMANDI PER LA GESTIONE DEI MESSAGGI

Sono riportati di seguito, i comandi per la gestione dei messaggi in EEPROM.

LETTURA NUMERO ULTIMO MESSAGGIO MEMORIZZABILE

Sequenza di Input:
Codice: ***77*** ***(4D Hex)***
Mnemonico: ***M***

Questo comando permette di acquisire il numero dell'ultimo messaggio memorizzabile nella EEPROM gestita dal **μP25**. Questo varia in funzione del dispositivo installato e corrisponde a quanto riportato nella seguente tabella.

EEPROM	N.MAX
24c02 (256 Bytes)	23
24c04 (512 Bytes)	48
24c16 (2048 Bytes)	202

FIGURA 10: NUMERO DELL'ULTIMO MESSAGGIO MEMORIZZABILE IN EEPROM

Codici di Riposta:
Il numero dell'ultimo messaggio memorizzabile viene restituito in due nibble; in particolare:

Nibble L: (MSB) 0 0 0 0 Bit 3 Bit 2 Bit 1 Bit 0 (LSB)
Nibble H: 0 0 0 0 Bit 7 Bit 6 Bit 5 Bit 4

RICHIESTA DI EEPROM PRONTA

Sequenza di Input:
Codice: ***66*** ***(42 Hex)***
Mnemonico: ***B***

Mediante questo comando é possibile richiedere al **μP25** se é pronto per gestire un nuovo messaggio in EEPROM, esso va eseguito tutte le volte che si deve inviare uno dei comandi descritti in seguito per la gestione dei messaggi.

Codici di Risposta:
Il codice restituito dal **μP25** a seguito di tale comando sono i seguenti:

0 *EEPROM non disponibile per gestire un nuovo messaggio*
1 *EEPROM disponibile per gestire un nuovo messaggio*

MEMORIZZAZIONE DI UN MESSAGGIO

Sequenza di Input:

Codice: ***69 <nib. L Mess.> <nib. H Mess.>***
< nib. L car. 0> <nib. H car. 0>…< nib. L car. 9> <nib. H car. 9>
(45 <nib. L Mess.> <nib. H Mess.>
< nib. L car. 0> <nib. H car. 0>…< nib. L car. 9> <nib. H car. 9> Hex)

Mnemonico: ***E ASCII(nib. L Mess.) ASCII(nib. H Mess.)***
ASCII(nib. L car. 0) ASCII(nib. H car. 0) …
… ASCII(nib. L car. 9) ASCII(nib. H car. 9)

Il messaggio, di 10 caratteri, con codice indicato in "Mess." viene memorizzato nella EEPROM. Il numero del messaggio , che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **0÷n.max**, dove "n.max" é il numero dell'ultimo messaggio memorizzabile, visto in precedenza, e ricavabile direttamente dalla tabella precedente o tramite l'apposito comando.

Car. x (nibble L) :	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
(nibble H) :		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	

Questi byte possono essere compresi in tutto il range **0÷255** (**0÷FF Hex**).

Note:
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:
Se si desidera memorizzare, come messaggio numero 16, la stringa "ABCDEFGHIJ" (corrispondente ai codici: 65, 66, 67, 68, 69, 70, 71, 72, 73, 74), sarà necessario inviare la seguente sequenza:
69 0 1 1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4.

LETTURA DI UN MESSAGGIO

Codice: ***76 <nib. L Mess.> <nib. H Mess.>***
(4C <nib. L Mess.> <nib. H Mess.> Hex)

Mnemonico: ***L ASCII(nib. L Mess.) ASCII(nib. H Mess.)***

Il messaggio, di 10 caratteri, con codice indicato in "Mess." viene letto dalla EEPROM e restituito in seriale.
Il numero del messaggio , che deve essere inviato a nibble secondo il formato indicato in precedenza, deve essere compreso nel range **0÷n.max**, dove "n.max" ha lo stesso significato visto in precedenza e ricavabile direttamente dalla tabella precedente o tramite l'apposito comando.

Codici di Risposta:
I codici dei 10 caratteri che compongono il messaggio sono restituiti a nibble secondo,il formato visto per il comando precedente.

Note:
Tale comando viene ignorato, nel caso che la relativa sequenza contenga dei dati non validi.

Esempio:
Facendo riferimento all'esempio del comando precedente, se si vuole leggere il messaggio numero 16 dalla EEPROM, sarà necessario inviare la seguente sequenza: **76 0 1**. La risposta del **µP25** a seguito del tale comando sarà la sequenza: **1 4 2 4 3 4 4 4 5 4 6 4 7 4 8 4 9 4 10 4**.

COMANDI RELATIVI ALLA GESTIONE DEL COUNTER A 16 BIT

Sono di seguito riportati i comandi relativi alla gestione del Counter a 16 bit del **µP25**. Tale contatore viene incrementato dalle variazioni di stato sul pin INT0, quando questo é configurato come segnale di Trigger, mediante il codice ed il comando apposito del modo SETUP.

LETTURA DEL CONTATORE A 16 BIT

Sequenza di Input:
Codice: ***73*** ***(49 Hex)***
Mnemonico: ***I***

Questo comando permette di acquisire il valore attuale del contatore a 16 bit del **µP25**.

Codici di Risposta:
La sequenza restituita a seguito del comando, è formata da 4 codici che esprimono la combinazione a 16 bit attualmente memorizzata all'interno dei registri del contatore; questa é trasmessa a nibble secondo il seguente formato:

Contatore	**(bit 0÷3)**	**:**	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
	(bit 4÷7)	**:**		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4
	(bit 8÷11)	**:**		0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8
	(bit 12÷15)	**:**		0	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12

Note:
Quando il contatore del **µP25** ha raggiunto il suo massimo valore, corrispondente a **65535** (**FFFF Hex**), ad un successivo impulso sul Trigger, il nuovo valore della combinazione sarà 0.
Se il pin INT0 é configurato come linea di Input o Output, a seguito di questo comando, sarà sempre restituita la combinazione **0**.

Esempio:
Se i registri del contatore a 16 bit contegono la combinazione 23055 (5A0F Hex), seguito della trasmissione del comando **73**, il **µP25** restituirà la seguente sequenza: **15 0 10 5**.

AZZERAMENTO DEL COUNTER A 16 BIT

Sequenza di Input:

Codice: ***88 120*** ***(58 78 Hex)***
Mnemonico: ***X x***

A seguito di questo comando, il **μP25** azzera i registri del counter a 16 bit, scrivendo in essi la combinazione **0**.

COMANDI RELATIVI ALLA GESTIONE DI A/D, D/A E M5480

Sono riportati di seguito relativi alla gestione, ad alto livello, dell'A/D converter **TCL 2543**, del D/A Converter **AD420** e del display driver **M5480**.
Tali dispositivi vanno interfacciati al **μP25**, sfruttando le varie linee del Port **2**, come illustato nel capitolo relativo alla descrizione hardware di tale collegamento.
Per poter sfruttare i vari comandi a disposizione, é neccessario attivare la gestione della periferica deriderata, tramite le apposite sequenze del modo SETUP,

CONVERSIONE SU UN CANALE DELL'A/D CONVERTER

Sequenza di Input:

Codice: ***81 <Canale>*** ***(51 <Canale> Hex)***
Mnemonico: ***Q ASCII(Canale)***

Esegue la conversione sul canale dell'A/D converter **TLC 2543**, interfacciato al **μP25**, tramite i Bit **0÷3** del Port **2**. Il numero del canale in questione, é indicato nel parametro "Canale". Tale byte deve essere compreso nel range di valori **0÷10**.
La combinazione restituita é espressa a 12 bit, quindi varia nel range **0÷4095** (**0÷3FF Hex**), in particolare si hanno le seguenti relazioni:

Combinazione = 0 -> $Vin = V_{ref-}$
Combinazione = 2048 (200 Hex) -> $Vin = (V_{ref+} - V_{ref-}) / 2$
Combinazione = 4095 (3FF Hex) -> $Vin = V_{ref+}$

Dove con **Vin** si intende la tensione applicata all'ingresso da convertire, mentre con $\mathbf{V_{ref+}}$ e $\mathbf{V_{ref-}}$ si intendondo, rispettivamente, il valore positivo e negativo della tensione di riferimento dell'A/D.

Codici di Risposta:
La combinazione a 12 bit viene restituita in a nibble, in 3 byte, secondo quanto riportato di seguito:

Combinazione	**(bit 0÷3)**	**:**	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
	(bit 4÷7)	**:**		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	
	(bit 8÷11)	**:**		0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8	

Note:
Se la gestione, ad alto livello, dell'A/D converter é disattivata, a seguito della richiesta di conversione, il **μP25** restituirà sempre il valore **0**. Se invece, la sequenza inviata contiene del codici non validi, il comando sarà ignorato.

Esempio:
Se si desidera eseguire una conversione sul canale 5 dell'A/D converter, nel quale é presente una tensione pari a V_{ref+}, sarà necessario inviare la seguente sequenza: **81 5**.
Dopo questa richiesta il **μP25** risponderà con i codici: **15 15 3**.

SETTAGGIO DELL'USCITA DEL D/A CONVERTER

Sequenza di Input:

Codice:	***113 <Bit 0÷3> <Bit 4÷7> <Bit 8÷11> <Bit 12÷15>***
	(71 <Bit 0÷3> <Bit 4÷7> <Bit 8÷11> <Bit 12÷15> Hex)
Mnemonico:	***q ASCII(Bit 0÷3) ASCII(Bit 4÷7) ASCII(Bit 8÷11) ASCII(Bit 12÷15)***

Setta l'uscita del D/A Converter **AD420**, interfacciato al **μP25** tramite i Bit **1, 6** e **7** del Port **2**, con il valore desiderato, della combinazione a **16 bit**.
Questa é inviata, a nibble, tramite i 4 parametri denominati "Bit x÷y", che hanno il seguente significato:

Combinazione	**(Bit 0÷3)**	**:**	(MSB)	0	0	0	0	Bit 3	Bit 2	Bit 1	Bit 0	(LSB)
	(Bit 4÷7)	**:**		0	0	0	0	Bit 7	Bit 6	Bit 5	Bit 4	
	(Bit 8÷11)	**:**		0	0	0	0	Bit 11	Bit 10	Bit 9	Bit 8	
	(Bit 12÷15)	**:**		0	0	0	0	Bit 15	Bit 14	Bit 13	Bit 12	

L'uscita del D/A converter sarà settata con un valore, in V o mA, proporzionale alla combinazione a 16 bit, relativa a tali parametri, in particolare:

Combinazione = 0	->	*Out = Minimo della scala settata*
Combinazione = 65535 (FFFF Hex)	->	*Out = Massimo della scala settata*

I valori minimo e massimo della scala, dipendono dal range settato sul D/A converter, tramite gli appositi pin, e possono essere: **0÷10 V**, **4÷20 mA**, **0÷20 mA** o **0÷24 mA**.

Note:
Se la gestione ad alto livello, del D/A converter é disattivata, il comando sarà ignorato; questo avverrà anche nel caso in cui la sequenza inviata contenga dei codici non validi.

Esempio:
Se si desidera settate la combinazione 23055 (5A0F Hex), sull'uscita del D/A converter, sarà necessario inviare al **μP25**, la seguente sequenza: **113 15 0 10 5**.

ATTIVAZIONE DELLE USCITE DEL LED DISPLAY DRIVER

Sequenza di Input:

Codice: ***109 <Out 1÷4> <Out 5÷8> <Out 9÷12>***
<Out 13÷16> <Out 17÷20> <Out 21÷23>
(6D <Out 1÷4> <Out 5÷8> <Out 9÷12>
<Out 13÷16> <Out 17÷20> <Out 21÷23> Hex)

Mnemonico: ***m ASCII(Out 1÷4) ASCII(Out 5÷8) ASCII(Out 9÷12)***
ASCII(Out 13÷16) ASCII(Out 17÷20) ASCII(Out 21÷23)

Setta, o resetta, le 23 uscite del LED display driver **M5480**, interfacciato al **µP25** tramite i Bit **4** e **5** del Port **2**, secondo quanto indicato tramite i byte “Out x÷y”.
Questi 6 parametri, infatti, permettono di stabilire quali linee devono essere attivate e quali disattivate, secondo la seguente corrispondenza:

Maschera uscite										
(Out 1÷4) :	(MSB)	0	0	0	0	Out 4	Out 3	Out 2	Out 1	(LSB)
(Out 5÷8) :		0	0	0	0	Out 8	Out 7	Out 6	Out 5	
(Out 9÷12) :		0	0	0	0	Out 12	Out 11	Out 10	Out 9	
(Out 10÷16):		0	0	0	0	Out 16	Out 15	Out 14	Out 13	
(Out 17÷20):		0	0	0	0	Out 20	Out 19	Out 18	Out 17	
(Out 21÷23):		0	0	0	0	0	Out 23	Out 22	Out 21	

Le 23 uscite del M5480 risulterannoattivate o disattivate a seconda dello stato dei vari bit che compongono la maschera, in particolare:

Out x	*0*	*Uscita x del M5480 DISATTIVATA*
	1	*Uscita x del M5480 ATTIVATA*

Note:
Se la gestione ad alto livello, del M5480 é disattivata, il comando sarà ignorato; questo avverrà anche nel caso in cui la sequenza inviata contenga dei codici non validi.

Esempio:
Se si desidera attivare le uscite 1,6, 8, 15, 22 e 23 del LED display driver M5480, sarà necessario inviare al **µP25**, la seguente sequenza: **109 1 10 0 4 0 6**.

LETTURA DEL BYTE DI CONFIGURAZIONE DI A/D, D/A E M5480

Sequenza di Input:

Codice: ***105 (69 Hex)***
Mnemonico: ***i***

Permette di acquisire il byte nel quale é indicato se sono attive o meno le gestioni, ad alto livello, dell'A/D converter **TLC2543**, del D/A converter **AD420** e del LED display driver **M5480**.

Codice di riposta:
Il byte restituito dal **µP25** a seguito di tale comando, ha il seguente significato:

Bit 0 *Gestione A/D converter TLC2543* ***On/Off***
Bit 1 *Gestione D/A converter AD420* ***On/Off***
Bit 2 *Gestione LED display driver M5480* ***On/Off***
Bit 3÷7 *Non utilizzati:* ***sono posti a 0***

I Bit 0, 1 e 2 di tale byte indicano se é attiva, o meno, la gestione del dispositivo relativo, in particolare:

Bit x *0* *Gestione dispositivo DISATTIVATA*
1 *Gestione dispositivo ATTIVATA*

Esempio:
Se é attiva la gestione ad alto livello dell'A/D converter e quella del M5480, a seguito del comando **105**, il **µP25** restituirà il codice **5**.

TABELLA RIASSUNTIVA DEI COMANDI DEL MODO RUN

Viene riportata di seguito la tabella riassuntiva dei comandi disponibili nel modo RUN.

COMANDO	CODICE	CODICE HEX	MNEMONICO
Scrittura Port	87 port nib.L nib.H	57 port nib.L nib.H	W ASCII(port) ASCII(nib.L) ASCII(nib.H)
Lettura Port	82 port	52 port	R ASCII(port)
SET Port.Bit	83 port bit	53 port bit	S ASCII(port) ASCII(bit)
CLEAR Port.Bit	67 port bit	43 port bit	C ASCII(port) ASCII(bit)
SET Port.Bit temporizzato	115 port bit nib.L nib.H	73 port bit nib.L nib.H	s ASCII(port) ASCII(bit) ASCII(nib.L) ASCII(nib.H)
CLEAR Port.Bit temporizzato	99 port bit nib.L nib.H	63 port bit nib.L nib.H	s ASCII(port) ASCII(bit) ASCII(nib.L) ASCII(nib.H)
Onda Quadra su Port.Bit	80 port bit nib.L nib.H	50 port bit nib.L nib.H	P ASCII(port) ASCII(bit) ASCII(nib.L) ASCII(nib.H)
Onda Quadra su Port.Bit temporizzata	112 port bit nib.L nib.H nib.L nib.H	70 port bit nib.L nib.H nib.L nib.H	p ASCII(port) ASCII(bit) ASCII(nib.L) ASCII(nib.H) ASCII(nib.L) ASCII(nib.H)
Lettura Port.Bit	114 port bit	72 port bit	r ASCII(port) ASCII(bit)
Lettura Port.Bit con Debauncing	68 port bit nib.L nib.H	44 port bit nib.L nib.H	D ASCII(port) ASCII(bit) ASCII(nib.L) ASCII(nib.H)
Azzeramento Counter 16 Bit	88 120	58 78	X x
Lettura Counter 16 Bit	73	49	I
Reset Generale	65 97	41 61	A a

FIGURA 11: TABELLA 1 DEI COMANDI RELATIVI AL MODO RUN

COMANDO	CODICE	CODICE HEX	MNEMONICO
Rischiesta scrittura in EEPROM	66	42	B
Lettura Numero ultimo messaggio	77	4D	M
Memorizzazione Messaggio	69 nib.L nib.H nib.L0 nib.H0 .... nib.L9 nib.H9	45 nib.L nib.H nib.L0 nib.H0 .... nib.L9 nib.H9	E ASCII(nib.L) ASCII(nib.H) ASCII(nib.L0) ASCII(nib.H0) .... ASCII(nib.L9) ASCII(nib.H9)
Lettura Messaggio	76 nib.L nib.H	4C nib.L nib.H	L ASCII(nib.L) ASCII(nib.H)
Lettura canale A/D converter	81 Canale	51 Canale	Q ASCII(Canale)
Settaggio uscita D/A converter	113 nib.L0 nib.H0 nib.L1 nib.H1	71 nib.L0 nib.H0 nib.L1 nib.H1	q ASCII(nib.L0) ASCII(nib.H0) ASCII(nib.L1) ASCII(nib.H1)
Settaggio 23 uscite M5480	109 nib.L0 nib.H0 nib.L1 nib.H1 nib.L2 nib.H2	6D nib.L0 nib.H0 nib.L1 nib.H1 nib.L2 nib.H2	m ASCII(nib.L0) ASCII(nib.H0) ASCII(nib.L1) ASCII(nib.H1) ASCII(nib.L2) ASCII(nib.H2)
Lettura config. Port	70	46	F
Lettura config. pin INT0	102	66	f
Lettura config. A/D, D/A e M5480	105	69	i

FIGURA 12: TABELLA 2 DEI COMANDI RELATIVI AL MODO RUN

MODALITA' DI COMUNICAZIONE MASTER-SLAVE A 9 BIT

La modalità Master-Slave, sfrutta la tecnica di comunicazione a 9 bit.
In particolare oltre agli 8 bit di dati, viene gestito un nono bit che serve a distinguere una chiamata, da parte della apparecchiatura "**Master**" ad una delle strutture "**Slave**", da un normale passaggio di informazioni tra il Master e il dispositivo attualmente selezionato.
Quando il nono bit é posto a 1, il byte di dati deve contenere il nome, o codice di identificazione, del dispositivo con il quale si vuole comunicare, mentre ponendo questo particolare bit a 0 é poi possibile prelevare o fornire informazioni, a tale dispositivo.

Nel caso particolare della comunicazione con il **µP25**, il codice di identificazione deve essere quello settato tramite il comandi di settaggio Nome relativo alla modalità di funzionamento SETUP.
Quando viene inviato questo byte (con il nono bit posto a 1), il **µP25**, si riconosce e si pone attesa della stringa contenente, dati o comandi (con il nono bit posto a 0); questa deve essere al massimo di **24 byte**.
In questa stringa, può esserci solo un comando che comporta la restituzione di una informazione in seriale da parte del **µP25**; se ve ne é un numero superiore, i restanti comandi di questo tipo verranno ignorati.
Tra la trasmissione di un carattere ed il successivo, deve passare un tempo inferiore al tempo di **Time-Out**, in quanto, trascorso questo ritardo, si considera finita la stringa di dati ed inizia la fase di risposta.
I tempi di Time-Out relativi ai vari Baud Rate sono indicati di seguito:

Baud Rate	***Time-Out***
38400 Baud	*550 µsec*
19200 Baud	*990 µsec*
9600 Baud	*1.54 msec*
4800 Baud	*3.08 msec*
2400 Baud	*6.105 msec*
1200 Baud	*12,1 msec*

Quando scade il tempo di Time-Out, inizia la sequenza di risposta; questa constiste in un byte contenente il codice di presenza **13** (**0D Hex**), oppure da un dato o sequenza di dati, relativi ad un comando di lettura inviato nella chiamata precedente.

Esempio:
Se viene trasmissa una strnga contenente il comando di lettura di un Port, si avrà che in quella particolare chiamata sarà restituito il codice di presenza, mentre nella successiva, verrà trasmesso il dato acquisito dal Port, richiesto in precedenza.

Dopo che é stata completata la trasmissione dell'ultimo carattere della stringa, sarà necessario attendere un tempo di:

"tempo di trasmissione di un carattere"+Time-Out

prima che arrivi il primo carattere della sequenza di risposta, trasmessa dal **µP25**.

Esempio:
Se si lavora a 38.4 KBaud, quando é stata completata la tramissione dell'ultimo carattere, é necessario attendere un tempo di circa 810 μsec, prima che sia completata la ricezione del primo dato di risposta trasmesso dal **μP25**.

Note:
Tra una chiamata e la successiva, é neccessario attendere un tempo che é funzione del numero di comandi inviati e del tipo di operazioni che questi comportano.

Una stringa di dati o comandi, trasmessa dal "Master" deve sempre contenere delle sequenze di comandi complete. Se una di queste risulta incompleta può venire ignorata, insieme alla sequenza di comando successiva, anche se completa.

DESCRIZIONE GENERALE DELL'ALB E25

L' **ALB E25** (**ABACO®** Link Bus Evaluable 25 linee di I/0), é una scheda valutativa, basata sul **μP25**, il quale é interfacciato ad una serie di pulsanti, LEDs e connettori, che permettono all'utente di verificare tutte le varie risorse di cui questo dispositivo é provvisto.
Tramite i pulsanti, infatti, é possibile testare, il funzionamento in Input delle 25 linee del **μP25**, mentre i LEDs permettoro di provare le varie modalità di Output. Tali segnali di I/O sono poi disponibili su due comodi connettori a scatolino a 20 vie, con pin-out normalizzato **ABACO® I/O**, che consentono di interfacciare le linee all'eventuale elettronica esterna sviluppata dall'utente.
Questo collegamento può venire facilitato utilizzando la serie di moduli **BLOCK**, del tipo **FBC**, che consentono di dipanare i vari Flat-Cable su comode morsettiere a rapida estrazione.
La scheda **ALB E25** é inoltre dotata di un doppio intefacciamento seriale, **RS232** e **RS485**, mediante il quale é possibile verificare le modalità di comunicazione **punto-punto** e **Master-Slave** del **μP25**.
Una ricca serie di programmi dimostrativi ed esempi di utilizzo, consentono un immediato utilizzo della scheda; detti programmi sono disponibili per il P.C. oltre che per tutte le numerose schede di CPU del vasto carteggio **ABACO®**.
Le principali caratteristiche dell'**ALB E25** sono riassunte di seguito:

- Scheda valutativa basata sul **μP25**.
- **25** pulsanti per la verifica delle modalità di Input delle linee del **μP25**.
- **25** LEDs di vario colore per la verifica delle modalità di Output delle linee del **μP25.**
- **2** connettori a scatolino a 20 vie, con pin-out normalizzato **ABACO® I/O**, per l'interfacciamento alle linee di I/O.
- **Jumper** per la selezione della modalità RUN/SETUP del **μP25**.
- **LED** di visualizzazione dello stato della linea /SETUP del **μP25**.
- EEPROM seriale da **512 byte** (24c04), per la memorizzazione di messaggi e configurazione del **μP25**; possibilità di espandere la capacità della EEPROM fino a **2KByte**.
- Linea seriale con intefacciamento in, **RS232** e **RS485**, selezionabile tramite jumper.
- Singola tensione di alimentazione di **+5 Vdc**.

SPECIFICHE TECNICHE DELL’ALB E25

CARATTERISTICHE GENERALI

CPU utilizzata:	87c51 con firmware **μP25**.
Numero linee di I/O:	24 linee dei Port 0, 1 e 2, intefacciate a LEDs, pulsanti e connettori.
Linee di conteggio:	Linea del pin INT0, intefacciata a LED, pulsante e conettore, da utilizzare, eventualmente, come 25-esima linea di I/O.
EEPROM di bordo:	512 byte (24c04) espandibile fino a 2KByte.
Oscillatore di bordo:	Quarzo da 14.7456 MHz.
Tipo di intefacciamento:	In RS232 o RS485, selezionabile tramite jumper.

CARATTERISTICHE FISICHE

Dimensioni:	155x72 mm.
Peso:	92 g.
Connettori:	CN1: Mosettieria 4 vie verticale. CN2: 20 vie scatolino verticale M. CN3: 20 vie scatolino verticale M.
Range di temperatura:	Da 0 a 70 gradi centigradi.
Umidità relativa:	20% fino a 90% (senza condensa).

CARATTERISTICHE ELETTRICHE

Tensione di alimentaz.:	+5 Vdc.
Corrente assorbita:	**30 mA** con tutti i LEDs spenti. **60 mA** con tutti i LEDs accesi.

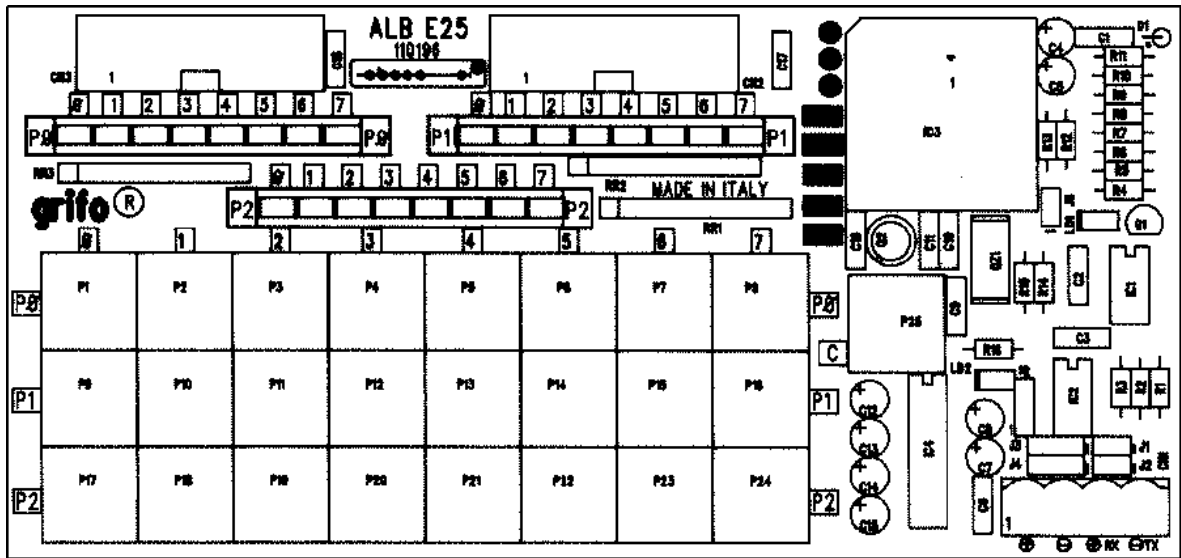



FIGURA 13: PIANTA COMPONENTI DELLA SCHEDA ALB E25

FIGURA 14: FOTO DELLA SCHEDA ALB E25

CONNESSIONI CON IL MONDO ESTERNO DELL'ALB E25

La scheda é provvista di 3 connettori con cui possono essere effettuati tutti i collegamenti delle linee di I/O con il mondo esterno. Di seguito viene riportata una loro descrizione comprensiva dei relativi pin-out; per una più facile individuazione di tali connettori, si faccia riferimento alla figura 18, mentre per ulteriori informazioni a riguardo del tipo di connessione, fare riferimento alle figure successive, che illustrano il tipo di collegamento effettuato a bordo scheda.

CN1 - CONNETTORE PER ALIMENTAZIONE E LINEA SERIALE

CN1 é un connettore a mosettiera da 4 vie, con il quale é possibile fornire alimentazione all'**ALB E25** ed effettuare il collegamento seriale. La morsettiera consentedi ancorare con estrema sicurezza tutti i cavi di sezione inferiore a 3 mm e di effettuare un comodo collegamento con il mondo esterno.

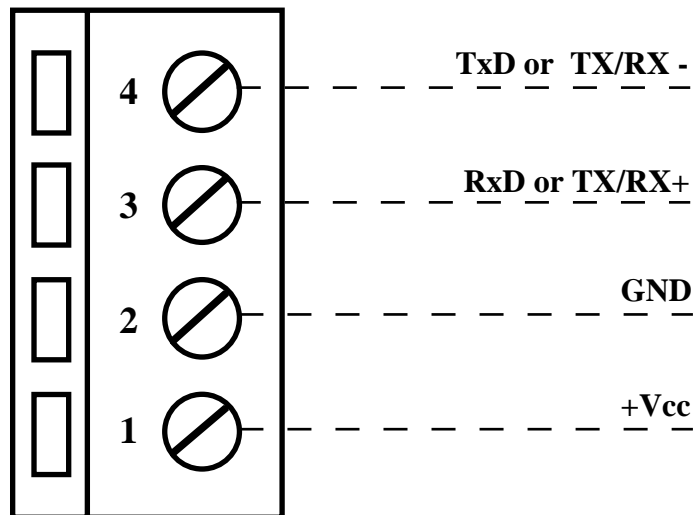



FIGURA 15: CONNETTORE CN1 DELL'ALB E25 PER ALIMENTAZIONE E LINEA SERIALE

Legenda:

GND	=		- Linea di massa per tensione di alimentazione e linea seriale.
+Vcc	=	I	- Linea di alimentazione a +5 Vdc.
TxD	=	O	- Transmit Data: linea di trasmissione in RS232.
RxD	=	I	- Receive Data: linea di ricezione in RS232.
TX/RX-	=	I/O	- Transmit/Receive Data Negativa: linea bipolare negativa per la trasmissione/ricezione seriale differenziale in RS485.
TX/RX+	=	I/O	- Transmit/Receive Data Positive: linea bipolare positiva per la trasmissione/ricezione seriale differenziale in RS485.

CN2 - CONNETTORE PER I/O DI PORT 2 E INT0

CN2 é un connettore a scatolino maschio, con passo 2.54 mm, da 20 piedini, tramite il quale si interfacciano le linee di I/O del port 2 e del pin INT0 del µ**P25**, con il mondo esterno. Il pin-out é quello normalizzato **ABACO® I/O**, ed i segnali logici presenti su tale connettore sono di tipo TTL.

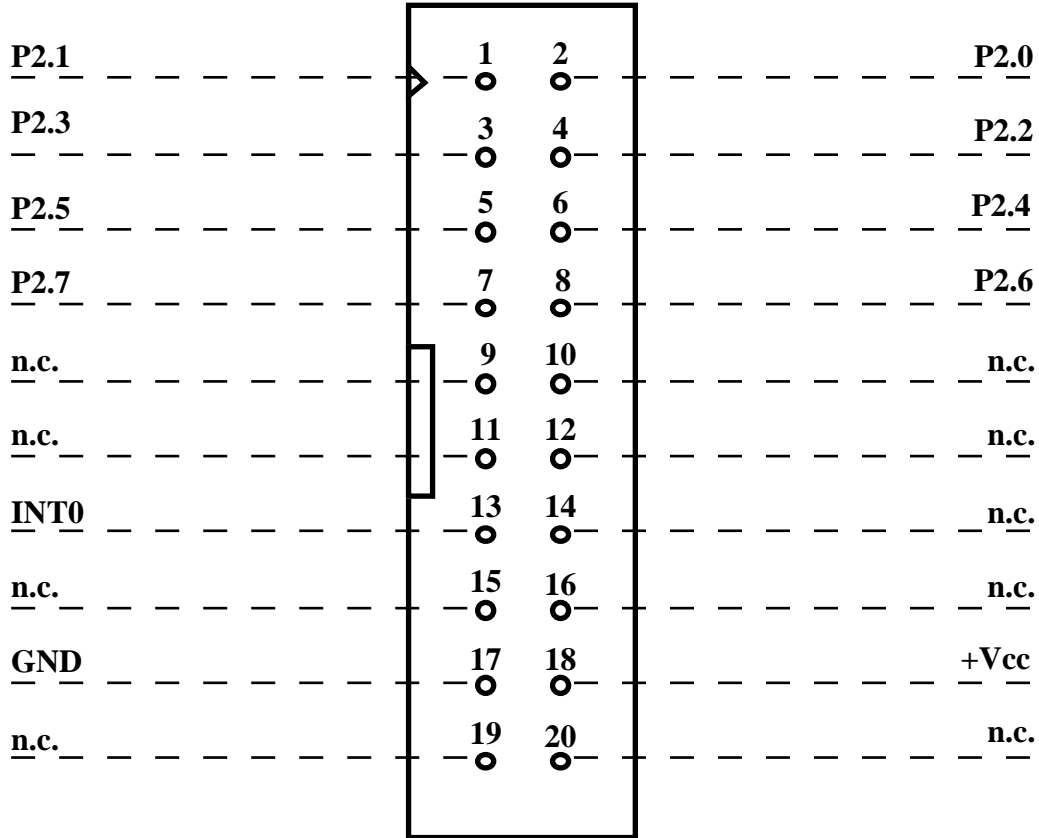



FIGURA 16: CONNETTORE CN2 DELL'ALB E25 PER I/O DI PORT 2 E INT0

Legenda:

GND	=		- Linea di massa di riferimento della scheda.
+Vcc	=	O	- Linea di alimentazione a +5Vcc.
P2.n	=	I/O	- Linea digitale n del port 2 del µ**P25**.
INT0	=	I/O	- Linea digitale del pin INT0 del µ**P25**.
n.c.	=		- Non collegato.

CN3 - CONNETTORE PER I/O DEI PORT 0 E 1

CN3 é un connettore a scatolino maschio, con passo 2.54 mm, da 20 piedini, tramite il quale si interfacciano le linee di I/O dei port 0 e 1 del **μP25**, con il mondo esterno. Il pin-out é quello normalizzato **ABACO® I/O**, ed i segnali logici presenti su tale connettore sono di tipo TTL.

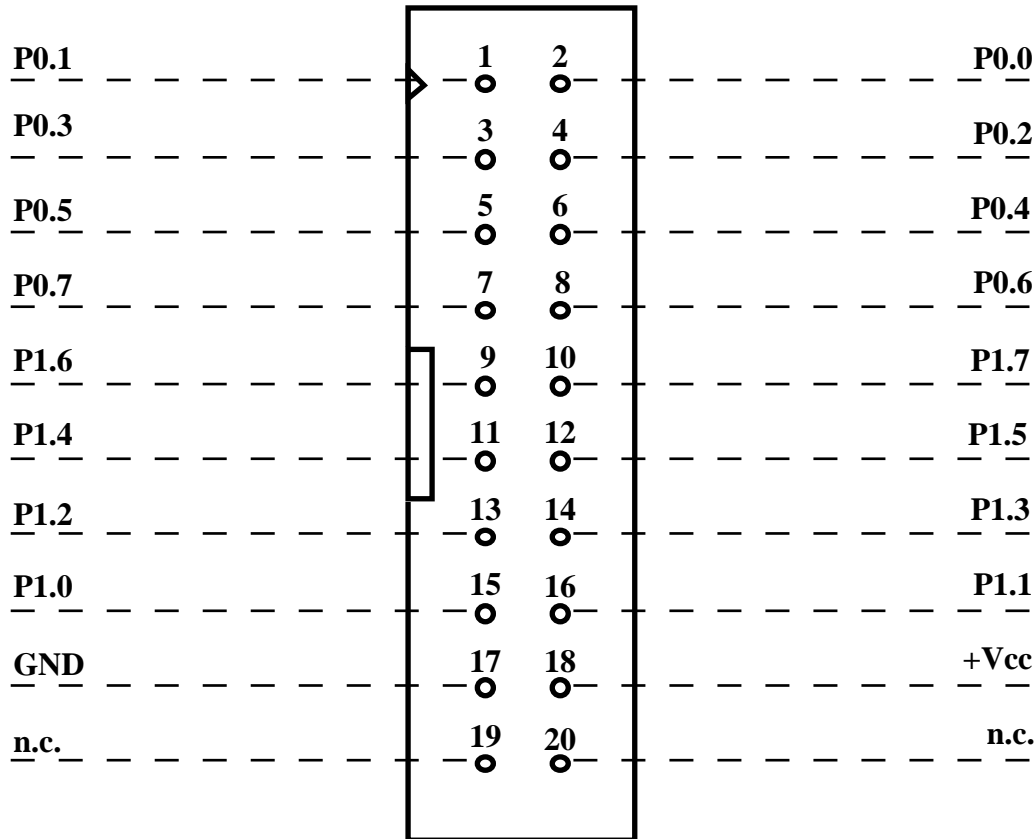



FIGURA 17: CONNETTORE CN3 DELL'ALB E25 PER I/O DEI PORT 0 E 1

Legenda:

GND	=		- Linea di massa di riferimento della scheda.
+Vcc	=	O	- Linea di alimentazione a +5Vcc.
P0.n	=	I/O	- Linea digitale n del port 0 del **μP25**.
P1.n	=	I/O	- Linea digitale n del port 1 del **μP25**.
n.c.	=		- Non collegato.

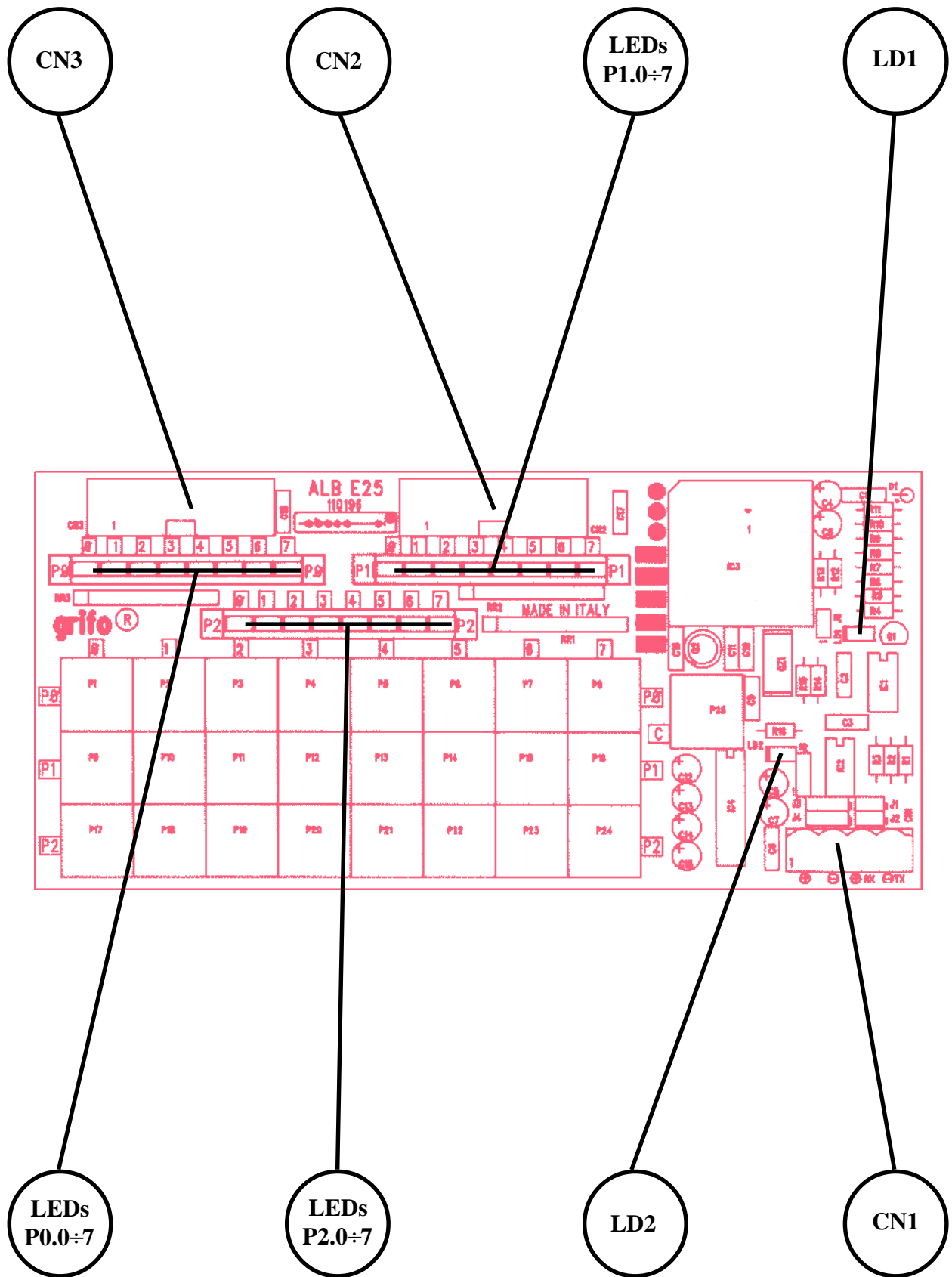



FIGURA 18: DISPOSIZIONE CONNETTORI E LEDS SULL'ALB E25

SEGNALAZIONI VISIVE DELL'ALB E25

La scheda **ALB E25** é dotata di 26 LEDs, di vario colore, con cui segnala lo stato delle linee di I/O e di /SETUP del **μP25**. Di seguito ne viene riportata una loro descrizione e per una più facile individuazione di tali componenti, si faccia riferimento alla figura 18.

LEDs	COLORE	FUNZIONE
LD1	Rosso	Visualizza lo stato della linea /SETUP del μP25. Il LED attivo corrsiponde al livello logico 0 di tale segnale.
LD2	Rosso	Visualizza lo stato della linea INT0 del μP25. Il LED attivo corrisponde al livello logico 0 di tale segnale.
P0.0÷P0.7	Rosso	Visulizzano, rispettivamente, lo stato delle 8 linee del port 0 del μP25. Il LED attivo corrisponde al livello logico 0 di relativo segnale.
P1.0÷P1.7	Giallo	Visulizzano, rispettivamente, lo stato delle 8 linee del port 1 del μP25. Il LED attivo corrisponde al livello logico 0 di relativo segnale.
P2.0÷P2.7	Verde	Visulizzano, rispettivamente, lo stato delle 8 linee del port 2 del μP25. Il LED attivo corrisponde al livello logico 0 di relativo segnale.

FIGURA 19: TABELLA DELLE SEGNALAZIONI VISIVE DELL'ALB E25

PULSANTI DELL'ALB E25

La scheda **ALB E25** é dotata di 25 pulsanti con i quali é possibile simulare la variazione di stato delle linee, settate in Input, del **μP25**. Ogni tasto corrisponde ad un contatto normalmente aperto, che si quando si chiude, pone a massa (livello logico **0**) la lonea collegata. In particolare vale la seguente corrispondenza:

Punsante premuto -> 0V = Livello logico **0**
Pulsante non premuto -> +5Vdc = Livello logico **1**

Di seguito ne viene riportata una descrizione dei vari pulsanti;per una facile individuazione di tali componenti, si faccia riferimento alla figura 20.

P1÷P8 -> Sono collegati, rispettivamente, alle linee P0.0÷P0.7 del **μP25**.
P9÷P16 -> Sono collegati, rispettivamente, alle linee P1.0÷P1.7 del **μP25**.
P17÷P24 -> Sono collegati, rispettivamente, alle linee P2.0÷P2.7 del **μP25**.
P25 -> E' collegato alla linea INT0 del **μP25**.

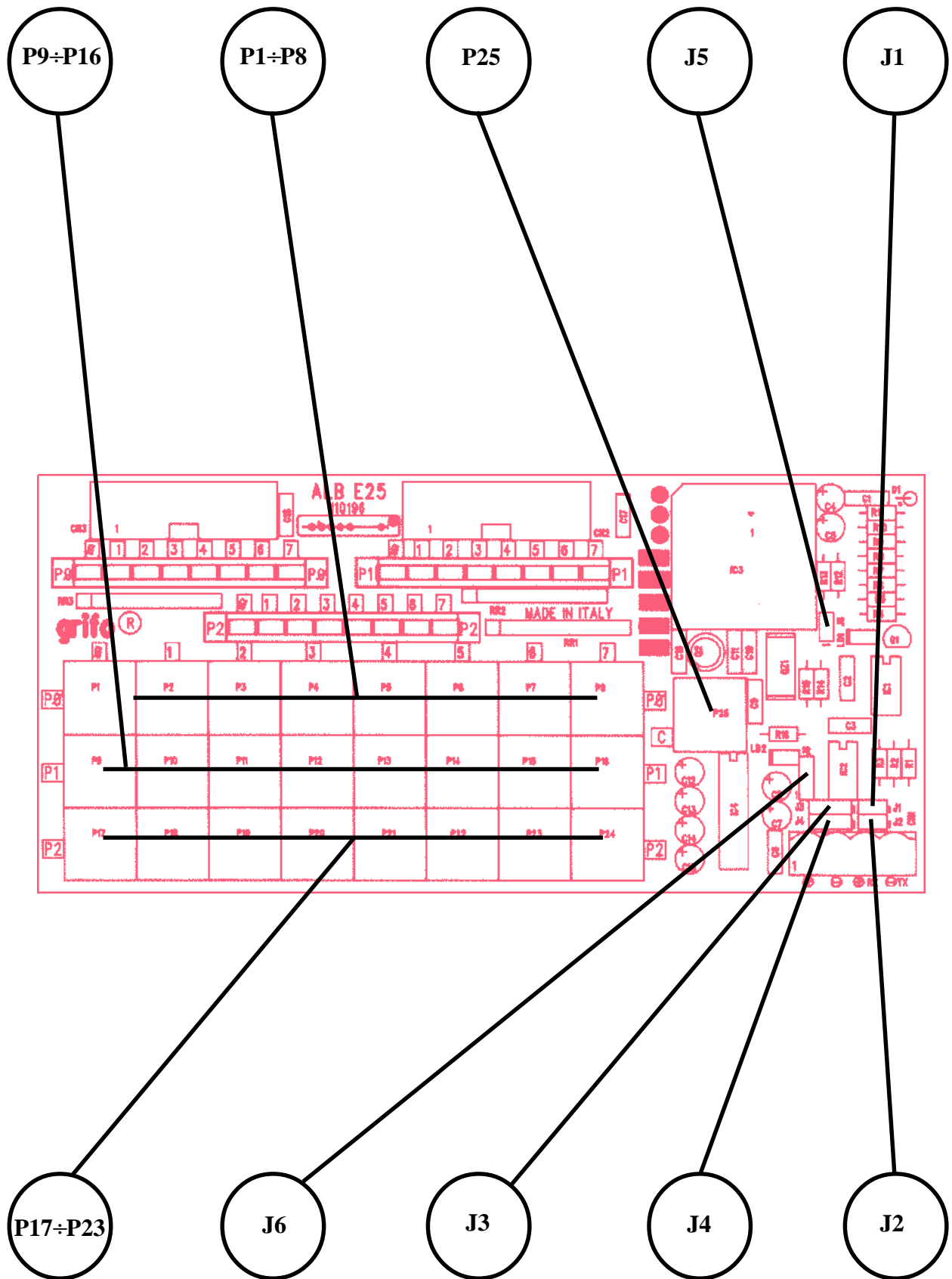



FIGURA 20: DISPOSIZIONE PULSANTI E JUMPERS SULL'ALB E25

JUMPERS DELL'ALB E25

Esistono a bordo della scheda **ALB E25**, 6 jumpers a cavaliere con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. In seguito ne é riportato l'elenco e la loro funzione nelle varie modalità di connessione.

JUMPERS	N. VIE	UTILIZZO
J1	2	In abbinamento al jumper J2, inserisce la circuiteria di terminazione sulla linea RS485.
J2	2	In abbinamento al jumper J1, inserisce la circuiteria di terminazione sulla linea RS485.
J3	3	In abbinamento ai jumpers J4 e J6, seleziona la comunicazione in RS232 o RS485.
J4	3	In abbinamento ai jumpers J3 e J6, seleziona la comunicazione in RS232 o RS485.
J5	2	Seleziona la modalità di funzionamento RUN o SETUP
J6	3	In abbinamento ai jumpers J3 e J4, seleziona la comunicazione in RS232 o RS485.

FIGURA 21: TABELLA RIASSUNTIVA DEI JUMPERS DELL'ALB E25

Di seguito é riportata una descrizione tabellare delle possibili connessioni dei 6 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa, oppure alla figura 13 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione di tali componenti a bordo della scheda, si utilizzi invece la figura 20 di questo manuale.

JUMPERS A 2 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J1	Non connesso	In abbinamento a J2 non collega la circuiteria di terminazione alla linea RS485.	
	Connesso	In abbinamento a J2 collega la circuiteria di terminazione alla linea RS485.	*
J2	Non connesso	In abbinamento a J2 non collega la circuiteria di terminazione alla linea RS485.	
	Connesso	In abbinamento a J2 collega la circuiteria di terminazione alla linea RS485.	*
J5	Non connesso	Seleziona la modalità di funzionamento RUN del µP25.	*
	Connesso	Seleziona la modalità di funzionamento SETUP del µP25.	

FIGURA 22: TABELLA DEI JUMPERS A 2 VIE DELL'ALB E25

Il simbolo ***** indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 3 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J3	posizione 1-2	In abbinamento ai jumpers J4 e J6, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J4 e J6, selezione la comunicazione in RS232.	*
J4	posizione 1-2	In abbinamento ai jumpers J3 e J6, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J3 e J6, selezione la comunicazione in RS232.	*
J6	posizione 1-2	In abbinamento ai jumpers J3 e J4, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J3 e J4, selezione la comunicazione in RS232.	*

FIGURA 23: TABELLA DEI JUMPERS A 3 VIE DELL'ALB E25

Il simbolo ***** indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

SELEZIONE DEL TIPO DI COMUNICAZIONE SERIALE DELL'ALB E25

La scheda **ALB E25** dipone della linea di comunicazione, che può essere bufferata in **RS232** oppure in **RS485**. La selezione di tale tipo di interfacciamento, avviene via hardware e viene effettuata tramite un'opportuna configurazione dei jumpers di bordo, come può essere desunto dalle precedenti tabelle. In particolare può essere:

J3, J4 e J6 in posizione 1-2 -> *Linea seriale configurata in RS485.*
J3, J4 e J6 in posizione 2-3 -> *Linea seriale configurata in RS232.*

N.B.
Tutte le restanti combinazioni ottenibili con tali jumper, non sono valide, e causano dei conflitti elettrici, che possono portare al danneggiamento della scheda stessa, oppure rovinare i driver di altri dispositivi collegati alla linea seriale.

Nel caso si utilizzi la linea seriale, in RS485, con i jumper J1 e J2 é possibile connettere la circuiteria di terminazione della linea differenziale di trasmissione/ricezione, formata dalle resistenze di forzatura (**R1=R3=3.3 KΩ**) e di carico della linea (**R2=120 Ω**). In particolare può essere:

J1 e J2 connessi -> *Circuiteria di terminazione Inserita.*
J1 e J2 non connessi -> *Circuiteria di terminazione Disinserita.*

DESCRIZIONE GENERALE DELL'ALB S25

L' **ALB S25** (**ABACO®** Link Bus Sperimentale 25 linee di I/0), é una scheda sperimentale, basata sul **µP25**, la quale é provvista di una vasta area prototipale in cui l'utente può sviluppare un proprio hardware da intefacciare alle linee di I/O, in modo da andare a specializzare la scheda in relazione alla propria applicazione.
Tale zona, inoltre é provvista di una serie di accorgimenti che consentono di facilitare il lavoro di prototipizzazione (alimentazioni in più punti, piazzole con fori maggiorati, ecc.).
Un'altra importante caratteristica dell'**ALB S25** é la possibilità di asportare quest'area prototipale, ottenendo un modulo dalle dimensioni molto contenute che, tramite due serie di strip passanti, può essere montato, in modalità *stack-through*, su di un back plane sviluppato dall'utente, nel quale é presente tutto l'hardware di interfacciamento con il campo.
La scheda **ALB E25** é inoltre dotata di un doppio intefacciamento seriale, **RS232** e **RS485**, mediante il quale é possibile verificare le modalità di comunicazione **punto-punto** e **Master-Slave** del **µP25**.
Una ricca serie di programmi dimostrativi ed esempi di utilizzo, consentono un immediato utilizzo della scheda; detti programmi sono disponibili per il P.C. oltre che per tutte le numerose schede di CPU del vasto carteggio **ABACO®**.
Le principali caratteristiche dell'**ALB S25** sono riassunte di seguito:

- Scheda sperimentale basata sul **µP25**.
- Area prototipale per lo sviluppo di un proprio hardware di specializzazione della scheda.
- Possibilità di rimuovere l'area prototipare e di installare il modulo rimanente, in modalità *stack-through*, su di un back plane sviluppato dall'utente.
- **Jumper** per la selezione della modalità RUN/SETUP del **µP25**.
- **LED** di visualizzazione dello stato della linea /SETUP del **µP25**.
- EEPROM seriale da **512 byte** (24c04), per la memorizzazione di messaggi e configurazione del **µP25**; possibilità di espandere la capacità della EEPROM fino a **2KByte**.
- Linea seriale con intefacciamento in, **RS232** e **RS485**, selezionabile tramite jumper.
- Singola tensione di alimentazione di **+5 Vdc**.

SPECIFICHE TECNICHE DELL'ALB S25

CARATTERISTICHE GENERALI

CPU utilizzata:	87c51 con firmware µ**P25**.
Numero linee di I/O:	24 linee dei Port 0, 1 e 2.
Linee di conteggio:	Linea del pin INT0 da utilizzare, eventualmente, come 25-esima linea di I/O.
EEPROM di bordo:	512 byte (24c04) espandibile fino a 2KByte.
Oscillatore di bordo:	Quarzo da 14.7456 MHz.
Tipo di intefacciamento:	In RS232 o RS485, selezionabile tramite jumper.

CARATTERISTICHE FISICHE

Dimensioni:	155x72 mm con area prototipale. 45x72 mm senza area prototipale.
Peso:	55 g con area prototipale.
Connettori:	CN1: Mosettieria 4 vie verticale. CN2: 28 vie predisposto per il montaggio di strip passanti.
Range di temperatura:	Da 0 a 70 gradi centigradi.
Umidità relativa:	20% fino a 90% (senza condensa).

CARATTERISTICHE ELETTRICHE

Tensione di alimentaz.:	+5 Vdc.
Corrente assorbita:	**35 mA**.

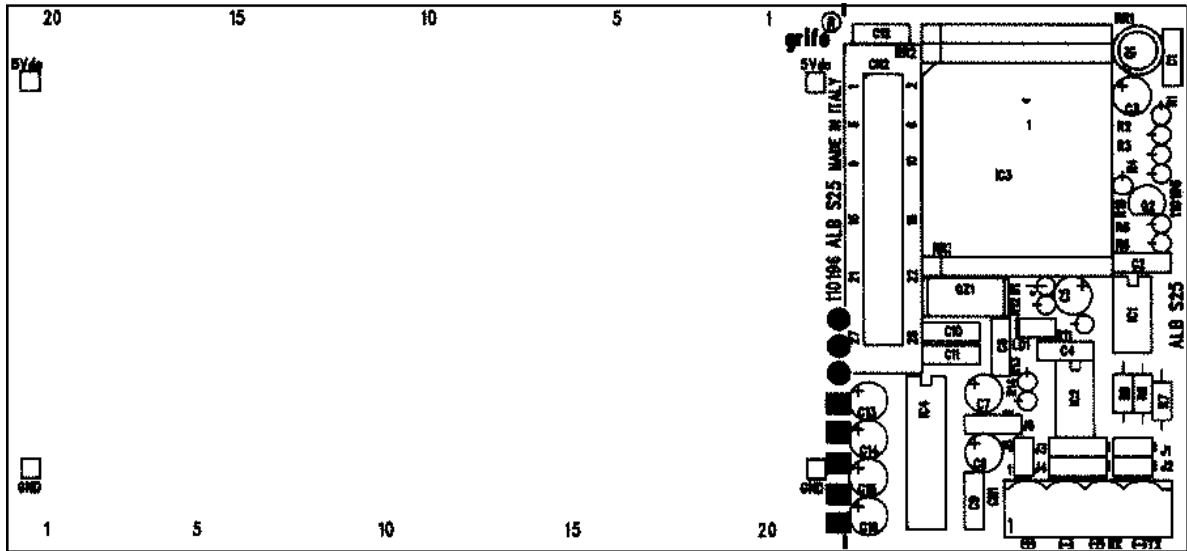



FIGURA 24: PIANTA COMPONENTI DELLA SCHEDA ALB S25

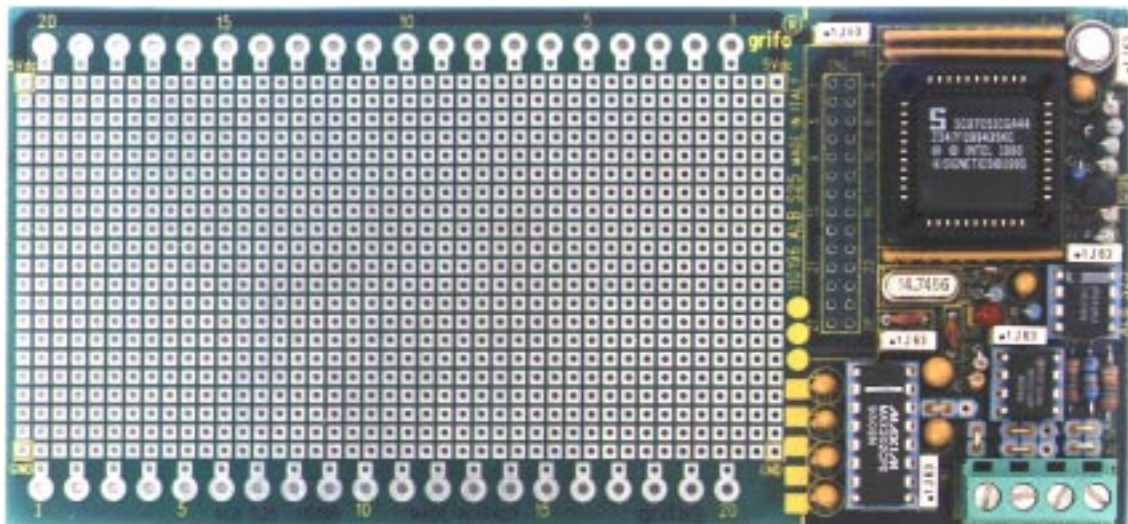



FIGURA 25: FOTO DELLA SCHEDA ALB S25

CONNESSIONI CON IL MONDO ESTERNO DELL'ALB S25

La scheda é provvista di 2 connettori con cui possono essere effettuati tutti i collegamenti delle linee di I/O l'hardware esterno. Di seguito viene riportata una loro descrizione comprensiva dei relativi pin-out; per una più facile individuazione di tali connettori, si faccia riferimento alla figura 27, mentre per ulteriori informazioni a riguardo del tipo di connessione, fare riferimento alle figure successive, che illustrano il tipo di collegamento effettuato a bordo scheda.

CN1 - CONNETTORE PER ALIMENTAZIONE E LINEA SERIALE

CN1 é un connettore a mosettiera da 4 vie, con il quale é possibile fornire alimentazione all'**ALB S25** ed effettuare il collegamento seriale. La morsettiera consentedi ancorare con estrema sicurezza tutti i cavi di sezione inferiore a 3 mm e di effettuare un comodo collegamento con il mondo esterno.

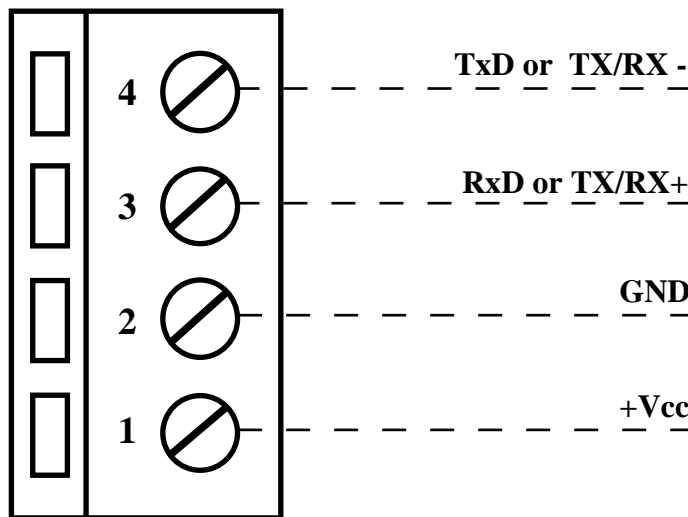



FIGURA 26: CONNETTORE CN1 DELL'ALB S25 PER ALIMENTAZIONE E LINEA SERIALE

Legenda:

GND	=		- Linea di massa per tensione di alimentazione e linea seriale.
+Vcc	=	I	- Linea di alimentazione a +5 Vdc.
TxD	=	O	- Transmit Data: linea di trasmissione in RS232.
RxD	=	I	- Receive Data: linea di ricezione in RS232.
TX/RX-	=	I/O	- Transmit/Receive Data Negativa: linea bipolare negativa per la trasmissione/ricezione seriale differenziale in RS485.
TX/RX+	=	I/O	- Transmit/Receive Data Positive: linea bipolare positiva per la trasmissione/ricezione seriale differenziale in RS485.

FIGURA 27: DISPOSIZIONE CONNETTORI, LEDS E JUMPERS SULL'ALB S25

CN2 - CONNETTORE PER I/O DI PORT E INT0

CN2 é un connettore, con passo 2.54 mm, da 28 piedini, tramite il quale si interfacciano le linee di I/O dei port e del pin INT0, con l'hardware esterno. I segnali logici presenti su tale connettore sono di tipo TTL; inoltre su CN2 é possibile montare 2 serie di strip passanti da 14 pin, per il montaggio in modalità *stack-through*, dell'**ALB S25** priva dell'area prototipale.

Segnale	Pin	Pin	Segnale
P0.1	1	2	P0.0
P0.3	3	4	P0.2
P0.5	5	6	P0.4
P0.7	7	8	P0.6
P1.6	9	10	P1.7
P1.4	11	12	P1.5
P1.2	13	14	P1.3
P1.0	15	16	P1.1
GND	17	18	+Vcc
P2.1	19	20	P2.0
P2.3	21	22	P2.2
P2.5	23	24	P2.4
P2.7	25	26	P2.6
INT0	27	28	SIP

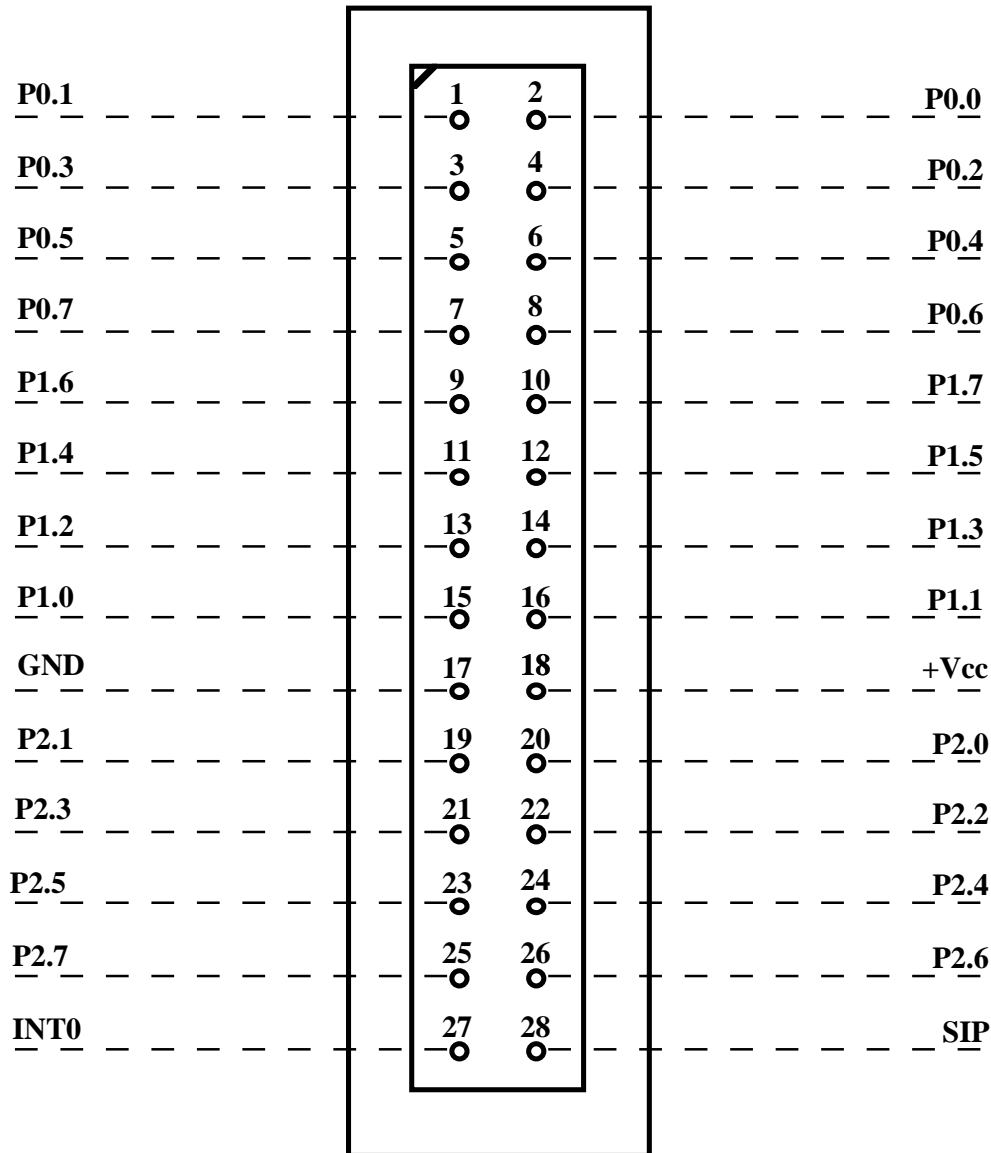



FIGURA 28: CONNETTORE CN2 DELL'ALB S25 PER I/O DI PORT E INT0

Legenda:

GND	=		- Linea di massa di riferimento della scheda.
+Vcc	=	I/O	- Linea di alimentazione a +5Vcc.
P0.n	=	I/O	- Linea digitale n del port 0 del **µP25**.
P1.n	=	I/O	- Linea digitale n del port 1 del **µP25**.
P2.n	=	I/O	- Linea digitale n del port 2 del **µP25**.
INT0	=	I/O	- Linea digitale del pin INT0 del **µP25**.
SIP	=	O	- Linea per l'alimentazione di reti resistive dell'hardware esterno.

JUMPERS DELL'ALB S25

Esistono a bordo della scheda **ALB S25**, 6 jumpers a cavaliere con cui é possibile effettuare alcune selezioni che riguardano il modo di funzionamento della stessa. In seguito ne é riportato l'elenco e la loro funzione nelle varie modalità di connessione.

JUMPERS	N. VIE	UTILIZZO
J1	2	In abbinamento al jumper J2, inserisce la circuiteria di terminazione sulla linea RS485.
J2	2	In abbinamento al jumper J1, inserisce la circuiteria di terminazione sulla linea RS485.
J3	3	In abbinamento ai jumpers J4 e J6, seleziona la comunicazione in RS232 o RS485.
J4	3	In abbinamento ai jumpers J3 e J6, seleziona la comunicazione in RS232 o RS485.
J5	2	Seleziona la modalità di funzionamento RUN o SETUP
J6	3	In abbinamento ai jumpers J3 e J4, seleziona la comunicazione in RS232 o RS485.

FIGURA 29: TABELLA RIASSUNTIVA DEI JUMPERS DELL'ALB S25

Di seguito é riportata una descrizione tabellare delle possibili connessioni dei 6 jumpers con la loro relativa funzione. Per riconoscere tali connessioni sulla scheda si faccia riferimento alla serigrafia della stessa, oppure alla figura 24 di questo manuale, dove viene riportata la numerazione dei pin dei jumpers, che coincide con quella utilizzata nella seguente descrizione. Per l'individuazione di tali componenti a bordo della scheda, si utilizzi invece la figura 27 di questo manuale.

JUMPERS A 2 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J1	Non connesso	In abbinamento a J2 non collega la circuiteria di terminazione alla linea RS485.	
	Connesso	In abbinamento a J2 collega la circuiteria di terminazione alla linea RS485.	*
J2	Non connesso	In abbinamento a J2 non collega la circuiteria di terminazione alla linea RS485.	
	Connesso	In abbinamento a J2 collega la circuiteria di terminazione alla linea RS485.	*
J5	Non connesso	Seleziona la modalità di funzionamento RUN del µP25.	*
	Connesso	Seleziona la modalità di funzionamento SETUP del µP25.	

FIGURA 30: TABELLA DEI JUMPERS A 2 VIE DELL'ALB S25

Il simbolo ***** indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

JUMPERS A 3 VIE

JUMPERS	CONNESSIONE	UTILIZZO	DEF.
J3	posizione 1-2	In abbinamento ai jumpers J4 e J6, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J4 e J6, selezione la comunicazione in RS232.	*
J4	posizione 1-2	In abbinamento ai jumpers J3 e J6, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J3 e J6, selezione la comunicazione in RS232.	*
J6	posizione 1-2	In abbinamento ai jumpers J3 e J4, selezione la comunicazione in RS485.	
	posizione 2-3	In abbinamento ai jumpers J3 e J4, selezione la comunicazione in RS232.	*

FIGURA 31: TABELLA DEI JUMPERS A 3 VIE DELL'ALB S25

Il simbolo ***** indica la connessione di default, ovvero la connessione impostata in fase di collaudo, con cui la scheda viene fornita.

SEGNALAZIONI VISIVE DELL’ALB S25

La scheda **ALB S25** é dotata di un unico LED, **LD1** di colore rosso, con cui segnala lo stato della linea /SETUP del **μP25**. In particolare, questo pin si trovelà a livello logico 0 in corrispondenza di ogni attivazione di LD1.
Per una più facile individuazione del componente in questione, si faccia riferimento alla figura 27.

SELEZIONE DEL TIPO DI COMUNICAZIONE SERIALE DELL’ALB S25

La scheda **ALB S25** dipone della linea di comunicazione, che può essere bufferata in **RS232** oppure in **RS485**. La selezione di tale tipo di interfacciamento, avviene via hardware e viene effettuata tramite un’opportuna configurazione dei jumpers di bordo, come può essere desunto dalle precedenti tabelle. In particolare può essere:

J3, J4 e J6 in posizione 1-2 -> *Linea seriale configurata in RS485.*
J3, J4 e J6 in posizione 2-3 -> *Linea seriale configurata in RS232.*

N.B.
Tutte le restanti combinazioni ottenibili con tali jumper, non sono valide, e causano dei conflitti elettrici, che possono portare al danneggiamento della scheda stessa, oppure rovinare i driver di altri dispositivi collegati alla linea seriale.

Nel caso si utilizzi la linea seriale, in RS485, con i jumper J1 e J2 é possibile connettere la circuiteria di terminazione della linea differenziale di trasmissione/ricezione, formata dalle resistenze di forzatura (**R7=R9=3.3 KΩ**) e di carico della linea (**R8=120 Ω**). In particolare può essere:

J1 e J2 connessi -> *Circuiteria di terminazione Inserita.*
J1 e J2 non connessi -> *Circuiteria di terminazione Disinserita.*

BIBLIOGRAFIA

E’ riportato di seguito, un elenco di manuali e note tecniche, a cui l’utente può fare riferimento per avere maggiori chiarimenti, sui vari componenti della struttura **μP25**, oppure montati a bordo delle schde **ALB E25** e **ALB S25**.

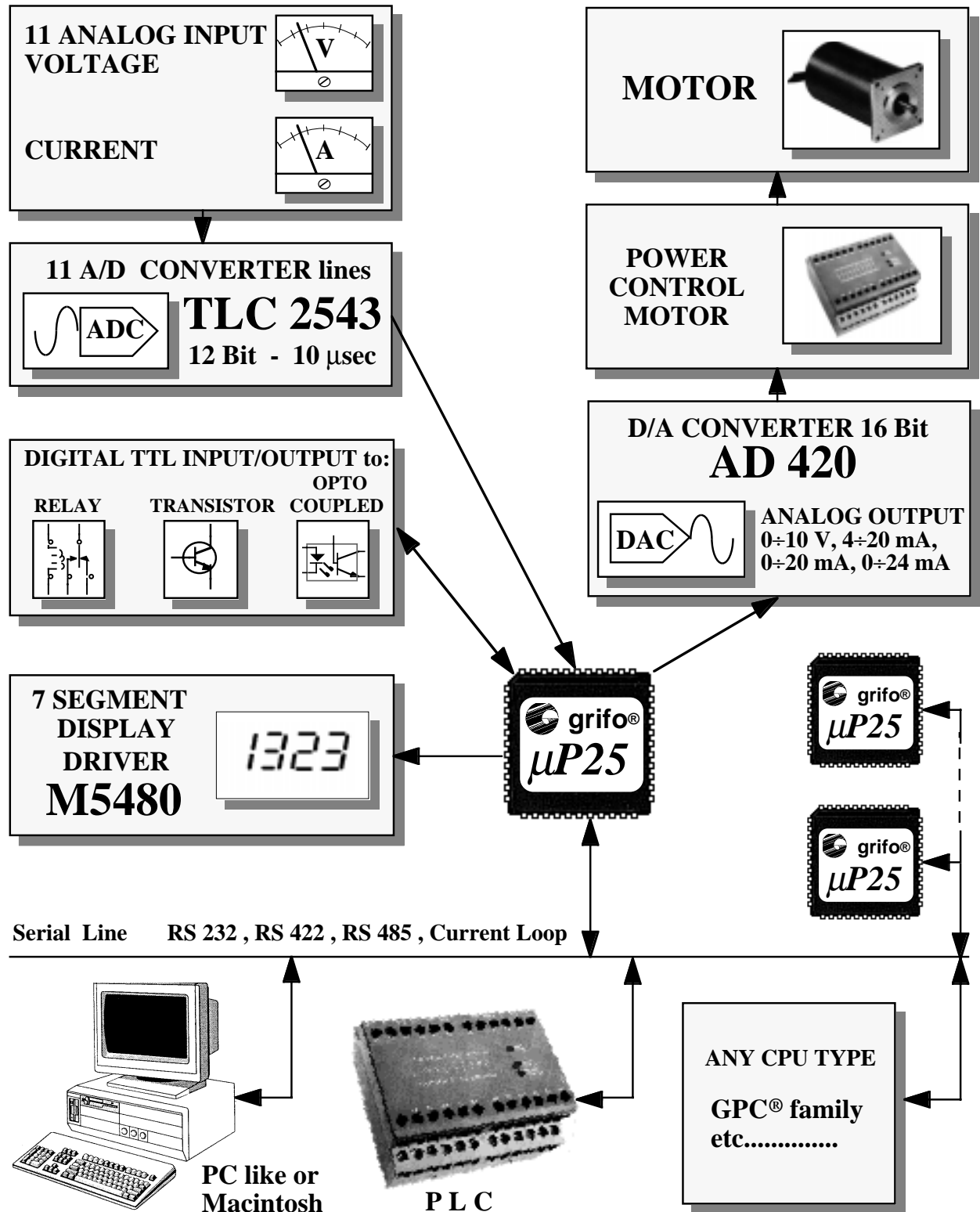
Manuale PHILIPS: *8051-based 8-bit Microcontrollers*
Manuale MAXIM: *Data Book - Volume III*
Manuale TEXAS INSTRUMENTS: *RS-422 and RS-485 Interface Circuits*
Manuale HEWLETT-PACKARD: *Optoelectronics Designer’s Catalog*
Manuale XICOR: *Data Book*
Manuale SGS-THOMSON: *Industrial and Computer Peripheral ICs*
Manuale NATIONAL SEMICONDUCTOR: *Interface Data Book*
Manuale ANALOG DEVICES: *Design-In Reference Manual - G1944-200-8/94*
Nota Tecnica TEXAS INSTRUMENTS: *TLC2543 12-bit Analog-to Digital Converter*
Application Report TEXAS INSTRUMENTS: *Microcontroller based Data Acquisition using the TLC2543 12-bit serial-out ADC*

µP25 Interconnections Blocks Diagram

FIGURA 32: SCHEMA DELLE STRUTTURE COLLEGABILI AL µP25

APPENDICE A: INDICE ANALITICO

D

E

F

G

I

J

L

segue L

M

O

P

R

S

segue S

T